2009-12

Development of star tracker system for accurate estimation of spacecraft attitude

Tappe, Jack A.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DEVELOPMENT OF STAR TRACKER SYSTEM FOR ACCURATE ESTIMATION OF SPACECRAFT ATTITUDE

by

Jack A. Tappe

December 2009

Thesis Co-Advisors: Jae Jun Kim
Brij N. Agrawal

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Development of Star Tracker System for Accurate Estimation of Spacecraft Attitude		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Jack A. Tappe			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

This thesis researches different star pattern recognition and attitude determination algorithms for a three-axis rotational spacecraft. A simulated star field will be suspended above the experimental Three-Axis Spacecraft simulator to provide a reference for the star-pattern recognition algorithms. A star field inertial reference frame database of stars will be developed with the simulator at zero attitude. The angle, planar triangle, and spherical triangle star pattern recognition algorithms will then be used to identify which stars are within a camera's field of vision. The imaged stars will then be matched up to the corresponding stars within the test bed database. With the imaged stars identified and the corresponding inertial frame reference data, the test bed's attitude will be determined using the least-squares, TRIAD, and Quaternion Estimator (QUEST) attitude determination algorithms. On the three-axis simulator, an iterative algorithm is developed to demonstrate increasing star tracker accuracy with each updated *A* matrix.

14. SUBJECT TERMS Star Tracker, Planar Triangles, Angle Method, Spherical Triangle, Quaternion Estimator, Least-Squares (QUEST), TRIAD			**15. NUMBER OF PAGES** 109
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

DEVELOPMENT OF STAR TRACKER SYSTEM FOR ACCURATE ESTIMATION OF SPACECRAFT ATTITUDE

Jack A. Tappe
Lieutenant, United States Navy
B.S., University of Idaho, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2009

Author: Jack A Tappe

Approved by: Jae Jun Kim
Thesis Advisor

Brij N. Agrawal
Co-Advisor

Dr. Knox T. Millsaps
Chairman, Department of Mechanical and Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis researches different star pattern recognition and attitude determination algorithms for a three-axis rotational spacecraft. A simulated star field will be suspended above the experimental Three-Axis Spacecraft simulator to provide a reference for the star-pattern recognition algorithms. A star field inertial reference frame database of stars will be developed with the simulator at zero attitude. The angle, planar triangle, and spherical triangle star pattern recognition algorithms will then be used to identify which stars are within a camera's field of vision. The imaged stars will then be matched up to the corresponding stars within the test bed database. With the imaged stars identified and the corresponding inertial frame reference data, the test bed's attitude will be determined using the least-squares, TRIAD, and Quaternion Estimator (QUEST) attitude determination algorithms. On the three-axis simulator, an iterative algorithm is developed to demonstrate increasing star tracker accuracy with each updated *A* matrix.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The objective of this research is the study of star trackers and the use of star pattern recognition algorithms and attitude determination algorithms by simulations and in a laboratory setting. The purpose of the algorithms is to resolve satellite attitude determination problems.

The intent is to analyze several star pattern recognition algorithms and determine the best algorithm in terms of accuracy and computing resources used. The algorithms used in the study are the angle algorithm, planar triangle algorithm, and the spherical triangle algorithms for use in star identification. All algorithms will be used to compare a reference or inertial image of stars to an image simulating a view seen by a star tracker onboard an orbiting satellite.

The star pattern recognition algorithms are an essential part in determining the attitude of a satellite in orbit. The star tracker's algorithms will match stars observed in the star tracker reference frame and calculate the unit vectors to those stars in the satellite's frame of reference. The algorithm will match the angles or triangles calculated by the star tracker in the body reference system to the same angles or triangles in the inertial reference system. Once the matches are made, the inertial star vectors are matched to the body-frame vectors. These vectors in the inertial and body-frame coordinated systems are then used in attitude determination algorithms to solve the problem that takes the form of $y = Ax$.

The attitude determination algorithms take the vectors to the stars in the inertial coordinate system and the vectors to the same stars as referenced in the body frame of the satellite to determine the direction cosine matrix of the satellite. Once the attitude matrix is determined, the quaternions or Euler angles of the satellite can be resolved. There are several algorithms in use for attitude determination, but three were studied in this report.

The algorithms studied are the least-squares, Quaternion-Estimator (QUEST), and the TRIAD algorithm. Each algorithm operates on unit vectors, but each has its differences regarding the mathematical approach. The least-squares algorithm uses the

pseudo-inverse operation in MATLAB. The QUEST algorithm performs eigenvalue calculations to determine the optimal quaternion. The TRIAD algorithm uses two vectors in the inertial coordinate system and the body frame to calculate the attitude determination matrix.

Star Tracker simulations will also be performed on a three-axis simulator, representing a spacecraft, in a laboratory environment. A camera will be installed on the simulator to act as a star tracker. A screen installed on the ceiling above the simulator will be used to simulate a star field. Using MATLAB, an inertial database of stars is created while the camera detects the stars much in the same way as a star tracker would. Since the stars are displayed in close proximity in the laboratory environment, iterative algorithm is developed to demonstrate star tracker with the test bed.

LIST OF ACRONYMS AND ABBREVIATIONS

ADCS	Attitude Determination Control System
CCD	Charge-Coupled Device
DCM	Direction Cosine Matrix
ECI	Earth-Centered Inertial
FOV	Field Of Vision
IMU	Inertial Measurement Unit
MST	Miniature Star Tracker
PSF	Point Spread Function
QUEST	Quaternion Estimator
SRDC	Spacecraft Research & Design Center
TAS-2	Three-Axis Simulator, Platform 2
TASS	Three-Axis Spacecraft Simulator

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank all the professors and other researchers at NPS who have assisted me so much with my studies here. I would like to especially thank Professors Barry Leonard, Brij Agrawal, Grand Master Shin, and Comrade Oleg Yakimenko (aka the Crazy Russian, the Cardinal, etc.). I would like to also thank Dan Sakoda for all his help with IDEA-S without which, the capstone design would not have been possible.

I would also like to single out Professor Jae Jun Kim my thesis advisor. Professor Kim's careful, slow, and generous time assisting me with my research and class projects made my learning so much easier. Without Professor Kim's guidance and assistance, this report would not have been possible.

I would also thank my family for all their kind help and understanding during this time. My wife Katrina's patience and understanding made my time here wonderful. My children Kyler, Eliza, and Jenna have been great helping out at the house and understanding when their parents need to go out. Great kids!

Lastly, I would like to thank the crew at the Trident Room for their help without which this research would not have been possible. I appreciate Linn and Loretto bringing all our orders up to the roof. I would like to thank Pete and Mai for all their support.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

Attitude determination is an extremely important aspect of any spacecraft especially a spacecraft required to point with some accuracy. Spacecraft attitude is defined as the alignment of the spacecraft's fixed body frame with respect to a specified frame. The spacecraft may need to point a camera or payload to satisfy a mission requirement, or the craft may need to position its solar arrays accurately at the sun to maintain its power requirements. Without accurate attitude determination, a "Lost in Space" condition could occur with the spacecraft tumbling uncontrollably in orbit. Therefore, an accurate method is needed for spacecraft attitude determination so the spacecraft can point its payload within the required accuracy.

In space, there are few "landmarks" available for attitude determination. The sun and Earth provide two reference points, but the spacecraft detector may not able to view these reference points. The Earth's magnetic field may be used for position determination, but this is a very inaccurate method. By far, the object in view for the majority of time is deep space and the stars.

The attitude problem can be solved by vector observations of stars by an onboard detector. For over 40 years, vector observation algorithms have been studied for solutions to satellite attitude determination problems (Weiss, Bar-Itzhack, & Oshman, 2005). The vector observations of stars via a satellite-mounted camera provide one input to the algorithm while the fixed inertial vector measurements of the same stars provide the reference. These observations, measurements, and algorithms are conducted by a satellite star tracker.

With the advent of more modern technology, the use of stars as a visual reference point has become more popular for spacecraft. Initially, star trackers required rough attitude estimations from other spacecraft sensors such as a sun sensor or magnetometers. Stars within view of the star tracker were locked onto and tracked. When the stars went out of sight, the coarse estimates were used until the algorithm was able to identify new

stars (Gwanghyeok & Junkins, 2003). All processing of stars via algorithms were conducted on the ground via telemetry. The attitude estimates were then uplinked to the craft. These initial trackers were very large, heavy, expensive, and extremely slow (Gwanghyeok & Junkins, 2003). That was in the past.

Space-qualified microcomputers now can compare camera observations to onboard star catalogs in firmware (Christian Liebe, 2002). Star trackers are now available as self-contained, autonomous units incorporated into the satellite's guidance system. The star trackers are now only one of many instruments that are available as a suite of attitude determination devices. The solutions from these attitude determination devices are fed into a filter to generate an overall description of the craft's attitude.

Star trackers are the most accurate method of determining spacecraft position especially when directing precise payloads such as energy weapons, laser communications, and optical payloads. Most star trackers are a Charge-Coupled Device (CCD) that positions the stars on the plane of the CCD. The CCD converts images into a digital signal, which makes this technology ideal for light detection.

There are many different algorithms for use with star trackers and for attitude determination results. The angle algorithm is the simplest method of matching unit vectors of stars in the inertial frame to stars observed in the body frame of the satellite, yet this algorithm is prone to erroneous matches. The spherical triangle algorithm is a more robust algorithm, but it requires more calculations and processing. The planar triangle algorithm is similar to the planar triangle method, but requires approximately the same amount of calculations as the planar triangle algorithm.

B. THESIS OVERVIEW

The work presented will demonstrate star tracker attitude determination for slewing maneuvers on a three-axis simulator. Algorithms will be developed in MATLAB to determine star identification in the body frame of the satellite and the inertial reference frame. The star identification algorithms will also calculate the star inertial unit vectors and body-frame unit vectors. Another topic studied will be the algorithms for matching inertial vectors to body-frame vectors, therefore determining the

attitude quaternions and Direction-Cosine-Matrices (DCMs) of the spacecraft. The aforementioned algorithms will be studied by simulations, as well as integration with other attitude determination devices on a test bed.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. SPACECRAFT ATTITUDE DETERMINATION DEVICES

There are many devices used today on modern spacecraft to accurately determine their position. Magnetometers can be used in conjunction with the Earth's magnetic field. Sun sensors can be used for attitude determination, but the sun must be visible to the spacecraft. The problem is that the sun sensor and magnetometers can only achieve an accuracy of 0.1 degree (Cole, 2004). A solution to the accuracy problem is using a star tracker.

Star trackers are the most accurate device in use for determining a spacecraft's position. The star tracker is essentially a camera for the sole purpose of observing star patterns as observed on the celestial sphere (Christian Liebe, 2002). The star tracker is attached to the satellite onboard computer as part of the ADCS (Attitude Determination and Control Subsystem) (Christian Liebe, 2002). The star tracker operates automatically, getting images of star patterns within its Field of Vision (FOV). The stars observed by the camera can then be identified and the orientation of the spacecraft can be calculated.

Modern star trackers are extremely sensitive cameras attached on the bus of the spacecraft. These star cameras are low mass, low power devices that can output centroids of stars or quaternions. The AeroAstro Miniature Star Tracker (MST) in Figure 1 is a Miniature Star Tracker with a mass of 375 grams (not including the baffle) that draws less than two Watts of power. The star tracker's accuracy is ± 70 arc-seconds on all three axes with a two Hz update rate. The star tracker is located away from any propulsion exhaust nozzles to prevent "blinding" the camera from propellant exhaust plumes.

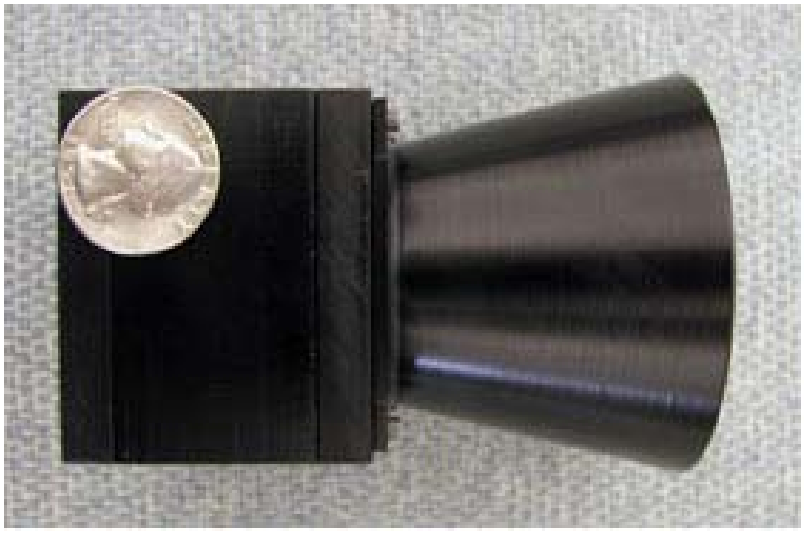

Figure 1. AeroAstro Miniature Star Tracker (From AeroAsto)

A star tracker is capable of attitude determination to within arc-second accuracy (Cole, 2004). For a high-cost spacecraft with accurate pointing requirements, the use of a star tracker is the preferred method for attitude determination. The aforementioned magnetometers and sun-sensors can be a back-up to the star trackers. Sky coverage is the percentage of the sky where the star tracker will operate.

Special considerations for star tracker construction must be taken into account to ensure the accuracy of the device. Each star tracker will have temperature limits at which the tracker must be maintained to ensure satisfactory operation. The star tracker will be thermally isolated from the spacecraft to minimize its sensitivity to fluctuations in the spacecraft temperature (The Aerospace Press, 2002). Shutters or other devices will ensure that the sun will never be able to shine down the tracker's optical boresight. The outer surface, like the one seen in Figure 1, will have a thermal finish to minimize effects of the space environment (The Aerospace Press, 2002). The aforementioned features and others ensure the optical-support structure is highly isothermal (The Aerospace Press, 2002).

Figure 2 is a representation of a generic star tracker, which illustrates the common components of a star tracker (Liebe, Dennison, Hancock, Stirbl, & Pain). The

microcomputer is the computer brain of the device and performs all necessary calculations (Liebe, Dennison, Hancock, Stribl, & Pain). The digitizer turns the image into a digital image, which is necessary for calculations. The microcomputer accesses the software star catalog to match the stars as visualized by the lens. The result or output from the device is the attitude quaternion with respect to the celestial sphere (Liebe, Dennison, Hancock, Stirbl, & Pain).

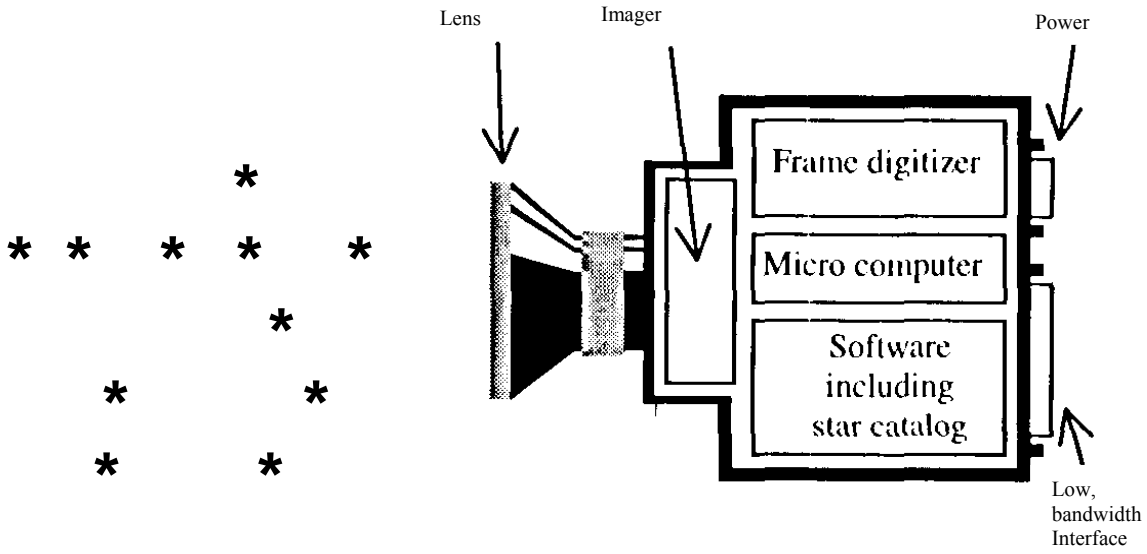

Figure 2. Star Tracker Diagram (From Liebe, Dennison, Hancock, Stirbl, & Pain).

To perform the calculations necessary for attitude determination, a set of reference stars and their inertial coordinates must be stored within the onboard database. Star Catalogs provide the necessary star information for programming a database of star data on a satellite.

B. STAR CATALOGS

The star catalog used for this research is the Hipparcos catalog of nearby stars. The Hipparcos catalog was obtained from the European Space Agency's Hipparcos astrometric mission that operated from November 1989 to March 1993 viewing the celestial sphere. The mission returned very high quality star astrometric and photometric data, specifically high precision data on 118,218 stars (Agency, 2009). The star catalog

is available for download from www.heasarc.gsfc.nasa.gov. Approximately half the Hipparcos star catalog was downloaded for use in the simulation.

Half of the Hipparcos catalog was used to illustrate the sky visible to a satellite in orbit. Each star has varies in magnitude and position on the celestial sphere. The Right Ascension and Declination of the stars were converted to **x** and **y** coordinates for plotting in two-dimensions using MATLAB. The stars are now an image the star tracker can view (Christian Liebe, 2002). Figure 3 is a plot of half the stars in the Hipparcos catalog. The star plot below will be discussed further.

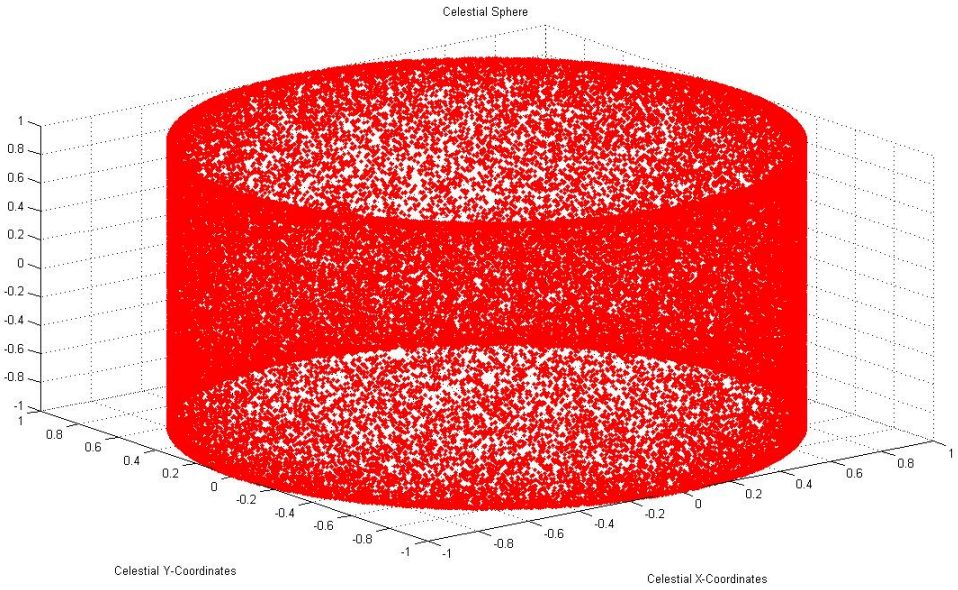

Figure 3. The Hipparcos star image.

C. STAR TRACKER OPERATION

Star trackers operate autonomously providing position data to the satellite ADCS system. Star trackers operate in two modes: 1) initial attitude determination and 2) attitude update determination (Christian Liebe, 2002). The first mode is the spacecraft not having any attitude knowledge, and the second mode is when the star tracker provides update attitude data to dated spacecraft attitude information. The second mode is also known as the tracking mode (Christian Liebe, 2002).

Stars in the star tracker FOV will be detected if their brightness is above the star tracker's minimum detection threshold. The brightness of the star relies upon the point spread function (PSF) as well as the star's position within the star tracker FOV (Christian Liebe, 2002). The PSF describes an imaging system's response to an impulse function or a point source. The star tracker above can detect fourth magnitude stars. Our Sun has an apparent magnitude of $M_V = -26.7$.

Star trackers calculate the centroids of the stars within the FOV by a process called centroiding (Christian Liebe, 2002). A star will appear to the star tracker as a source of light covering several pixels, due to lack of focus or smearing of the image due to satellite rotation. The centroid calculation will result in a (x_{cm}, y_{cm}) and a star intensity. Figure 4 illustrates stars as seen from the camera of the star tracker

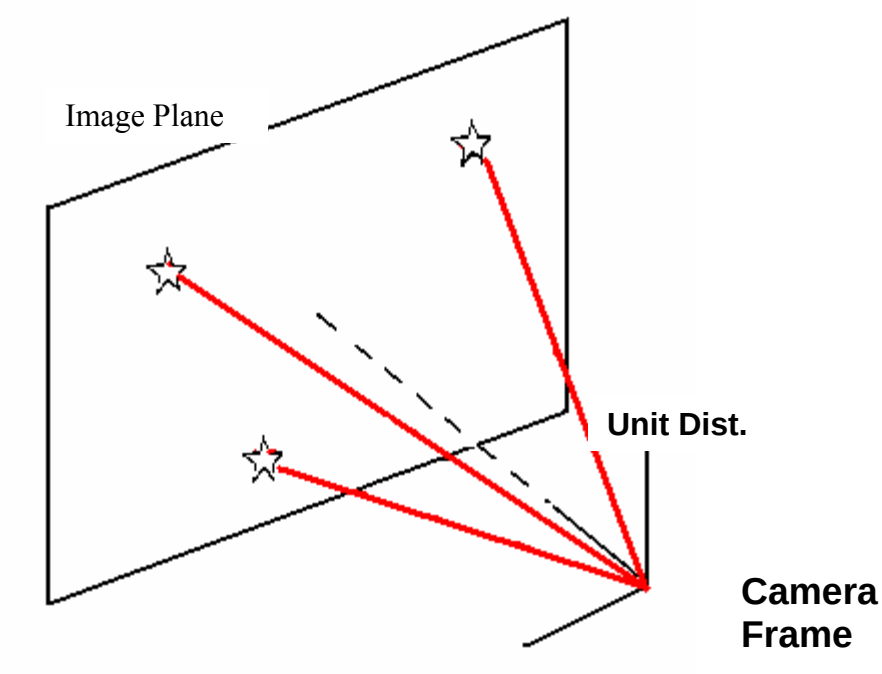

Figure 4. Stars as observed from the star tracker camera frame (From Diaz, 2006).

The mathematical formula for transferring the centroid coordinates into unit vectors according the star tracker body from are given as Equation 1 (Christian Liebe, 2002):

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} -X_C - X_o \\ -Y_C - Y_o \\ -f \end{pmatrix} \tag{1}$$

where (x, y) are the coordinates of the focal plane, (x_o, y_o) is the intersection of the focal plane and the optical axis, and F is the focal length of the camera. The star unit vectors, as referenced from an inertial coordinate system, are obtained via a star catalog.

The star tracker uses algorithms to match stars within its field of vision to the spacecraft's database of inertial star positions. The star tracker will update the position of the spacecraft at some frequency determined by the speed of the algorithm parsing through the stars within the FOV, and matching them to the onboard inertial database. The type of algorithm and the speed of the processing the database will determine the frequency of the star tracker attitude solutions. Naturally, the algorithm must also be able to match the stars even with sensor noise and smearing of the stars when the snapshot is taken.

D. STAR PATTERN RECOGNITION ALGORITHMS

1. Star Pattern Recognition Algorithms

The star tracker's star vectors, imaged and determined by the star tracker attached to the body frame of the satellite, are useless without some method to correlate them to a database of inertial star unit vectors. The matching of stars is accomplished by a matching algorithm. The algorithms are of the *lost-in-space* type or the *recursive* type, which runs off of some prior position knowledge (Spratling & Mortari, 2009).

Attitude calculations can occur only after stars are observed by the spacecraft. The star image must then be processed into data that is usable for the various algorithms in use by the star tracker. The algorithm then processes the image into usable body frame data. This image data must then be matched to an onboard database of star data. When matches occur, then the body-frame data is matched to its corresponding inertial data. The difference between the inertial data and the body-frame data gives the attitude of the

spacecraft. Subsequent algorithms covered in this report are used to establish a transformation that maps the observed star body vectors to the correct reference inertial vectors. Figure 5 illustrates the star pattern recognition, algorithm processing, and inertial database matching (Spratling & Mortari, 2009).

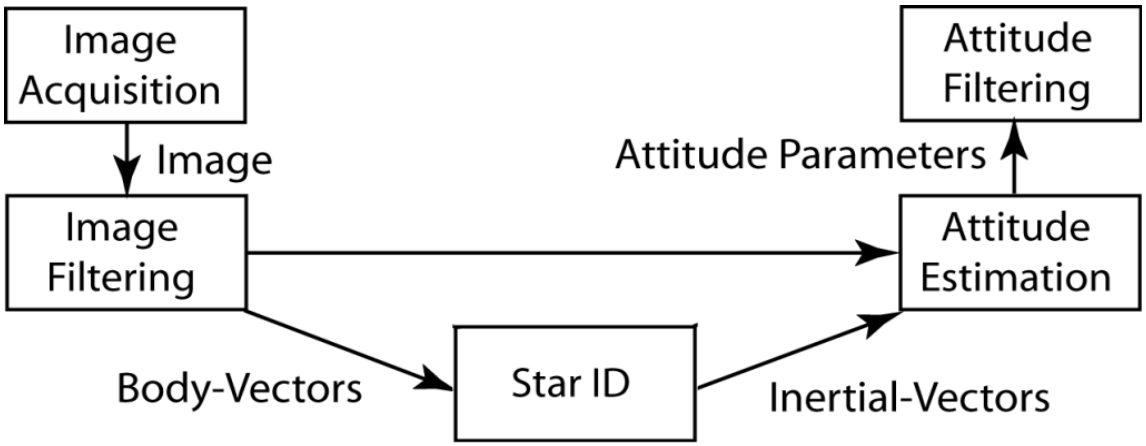

Figure 5. Star-Patter recognition algorithm flow (From Spratling & Mortari, 2009).

There are three basic pieces of each star pattern recognition algorithm. The first step is to extract some features from a set of star body vectors and brightness from stars within the trackers FOV (Spratling & Mortari, 2009). Next, the obtained information must be matched to entries within an onboard database to obtain the satellite's position. Finally, an estimate of the accuracy of this position is obtained (Spratling & Mortari, 2009).

Most of these algorithms involve calculations of angles between observed stars (Spratling & Mortari, 2009). The star pattern algorithms studied in this paper are the simple Angle Method, the Planar Triangles Method, and the Spherical Triangles algorithm. All of the above algorithms follow a few basic steps, which will be discussed further in this paper. The following is the generic sequence that assumes using the angle method for star pattern recognition:

1. Supply data – At some time, t_o, an image is obtained with visible stars. The tracker generates a tracker-framed position vector for each star (Needelman, Li, & Wu, 2005).

2. Select a primary pair – Two stars are selected and the angle is calculated between them.
3. Determine primary pair candidates – The separation angle between the two stars lies in a range of $\left[\theta - 3\varepsilon_{separation}, \theta + 3\varepsilon_{separation}\right]$, where $\varepsilon_{separation}$ is the one sigma error in the angle. If this pair corresponds to an angle in the lookup table, then obtain a set of inertial candidate stars, which are designated as candidate stars, C_{i1} and C_{i2} (Needelman, Li, & Wu, 2005).
4. Formulate primary assumption – Assume that the stars imaged by the star tracker, primary candidates, correspond to C_{i1} and C_{i2} which is termed the "primary assumption" (Needelman, Li, & Wu, 2005). If there are no matches, proceed to step 8.
5. Determine mapping between frames – Using the primary assumption above, find the direction cosine matrix $\boldsymbol{A}(t_o)$.
6. Apply Direct Match Test – Given $\boldsymbol{A}(t_o)$, you can now predict which entries in the database represent stars within the star tracker's view at time t_o (Needelman, Li, & Wu, 2005). If the above is valid, then the algorithm is terminated (Needelman, Li, & Wu, 2005). Otherwise, proceed to the next step.
7. New Primary Pair Candidate – Switch C_{i1}'s and C_{i2}'s position in the database and reprocess from step 4. Otherwise, proceed to step 8.
8. New Primary Pair – The two initial stars do not match any angles within the database. This may be due to errors in measurements or other reasons. Another pair is selected and another angle is calculated where the process begins at step 3 (Needelman, Li, & Wu, 2005).
9. Naturally, there are strengths and weaknesses of each algorithm.

a. Angle Algorithms

The angle method is the simplest star identification algorithm. Star pairs are observed by the camera, and their unit vectors are developed in reference to the frame of the star tracker, as shown in Figure 6. The star tracker then calculates the angles between all stars within the FOV of the camera as illustrated by Figure 10. The angle is calculated by the equation:

$$\theta = \cos^{-1}\left(r_1 \bullet r_2\right) \tag{2}$$

where $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are the unit vectors pointing to each star (Cole & Crassidus, 2006). The angle θ in Equation 2 will be the same from the inertial frame as it is viewed from the satellite. Figure 6 illustrates two stars detected by the star tracker in the camera frame with their respective x and y positions (Diaz, 2006).

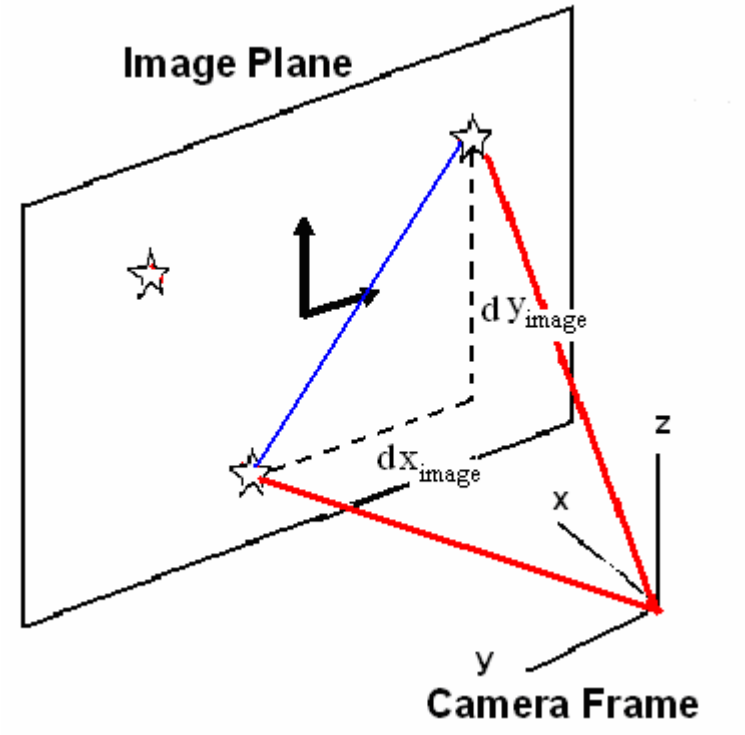

Figure 6. Angle detected calculated by star tracker (From Diaz, 2006).

The angle of the stars in the cameras FOV are calculated using Equation 2. However, the angle calculated is in the frame of the star tracker camera. The angles must be converted to the body frame of the satellite for use in any attitude determination algorithm. Those body frame angles must be compared to angles in the inertial reference frame. Therefore, an onboard database of inertial stars with their angles calculated from the inertial reference frame must be available. Figure 7 illustrates two stars with the angle between them as viewed from an inertial reference frame.

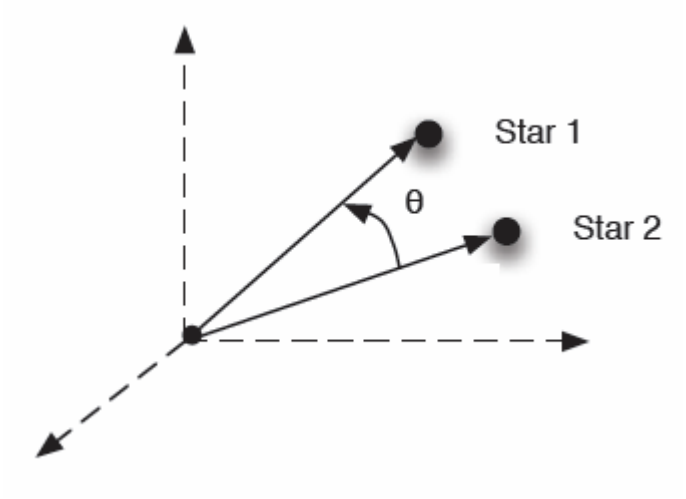

Figure 7. Angle method star pattern recognition algorithm (From Diaz, 2006).

The angle method of Figure 7 is simplistic and ideal for attitude determination, but there are significant drawbacks. The computational requirements for determing θ in Equation 2 are very simplistic. The angle measurement, however, will have a significant amount of error in it due to noise. This noise cannot be ignored in the calculations and must be dealt with. If the noise follows a Gaussian distribution, then standard deviation can be used to determine a range which these measurements will fall in (Cole & Crassidus, 2006). Also, care must be taken to ensure the two stars in the body frame of the satellite are correctly matched to their corresponding inertial stars, therefore careful logic must be setup to ensure correct star matching occurs.

b. Planar Triangles Algorithm

Another algorithm for star identification is the method of planar triangles. The star tracker develops a triangle from a combination of three stars as shown in Figure 8. The benefit of this algorithm is that more information can be obtained from a triangle than an angle, which will allow the star tracker to determine the satellite's attitude faster and use fewer stars than the aforementioned angle algorithm (Cole & Crassidus, Fast Star-Pattern Recognition Using Planar Triangles, 2006).

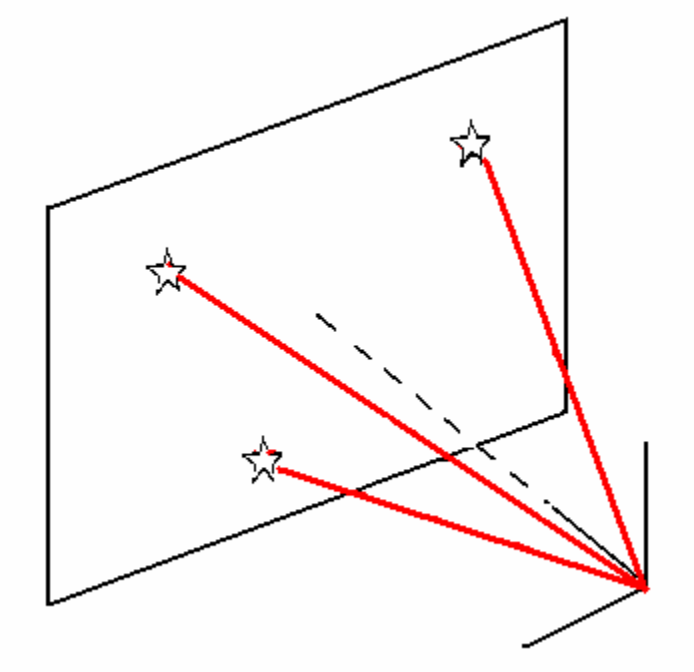

Figure 8. Three stars as viewed by the star tracker (From Diaz, 2006).

From the calculated triangle, the triangles area and polar moment can be determined. The area and polar moment provide two pieces of information vice the single angle developed by the angle algorithm. By observing three stars with unit vectors $\tilde{b}_1$, $\tilde{b}_2$, and $\tilde{b}_3$, the area of the planar triangle is obtained by Heron's formula (Cole & Crassidus, Fast Star-Pattern Recognition Using Planar Triangles, 2006):

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{3}$$

where

$$s = \frac{1}{2}(a+b+c) \tag{4}$$

$$a = \left\| \tilde{b}_1 - \tilde{b}_2 \right\| \tag{5}$$

$$b = \left\| \tilde{b}_2 - \tilde{b}_3 \right\| \tag{6}$$

$$c = \left\| \tilde{b}_1 - \tilde{b}_3 \right\| \tag{7}$$

Equations 3, 4, 5, 6, and 7 are valid in the Earth-Centered-Inertial or ECI frame as well as the star tracker frame (Cole & Crassidus, Fast Star-Pattern Recognition Using Planar Triangles, 2006). In the planar triangle method, three observed stars in Figure 9 provide

far more information than only two stars using the angle method. As shown, there are multiple angle calculations as well as other features of the triangle to store.

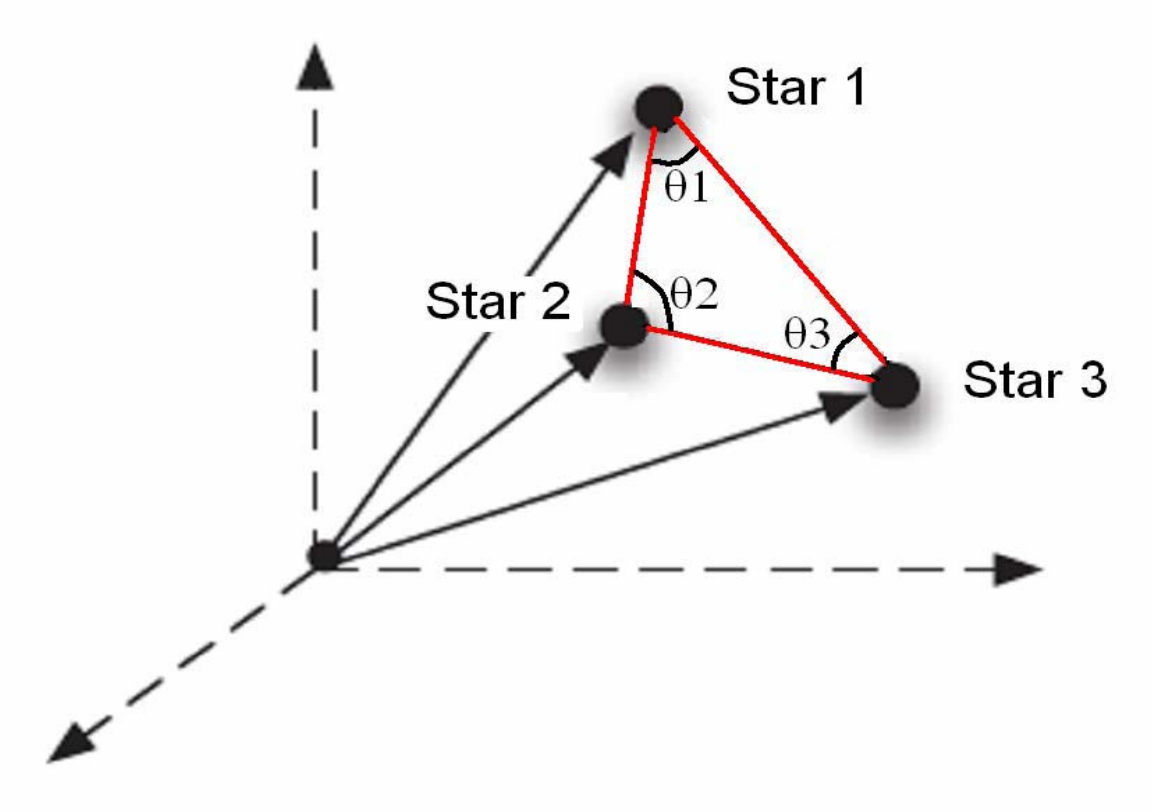

Figure 9. Star Tracker body unit vectors obtained for planar triangles From Diaz, 2006).

It will also be necessary to obtain the polar moment in conjunction with the area of the triangle. Two triangles may have the same area, but will have different second moments (Cole & Crassidus, Fast Star-Pattern Recognition Using Planar Triangles, 2006). The polar moment is calculated using Equations 3, 5, 6, and 7, in addition to:

$$J = A\left(a^2 + b^2 + c^2\right) \Big/ 36 \tag{8}$$

When using planar triangles, the use of the triangles polar moment and planar area will rapidly reduce the number of similar solutions, however there are certain costs with using this algorithm. There are significantly more features a triangle can provide when compared to an angle. Naturally, instead of determining the satellites

position with two stars, it now requires three stars if using the planar triangle algorithm. There are, however, significantly more mathematical calculations that must be performed when using the triangle algorithm compared to the angle calculations. Also, with the triangle there are two data points for each triangle which will require a larger memory to hold this data.

c. Spherical Triangles Algorithm

As with the planar triangle method, a similar algorithm used in star trackers is the use of spherical triangles. The star tracker calculates a spherical triangle when it observes three stars within its FOV, as in Figure 10. Again, the polar moment and area are used to determine which spherical triangle is being observed by the star tracker (Cole & Crassidus, 2004).

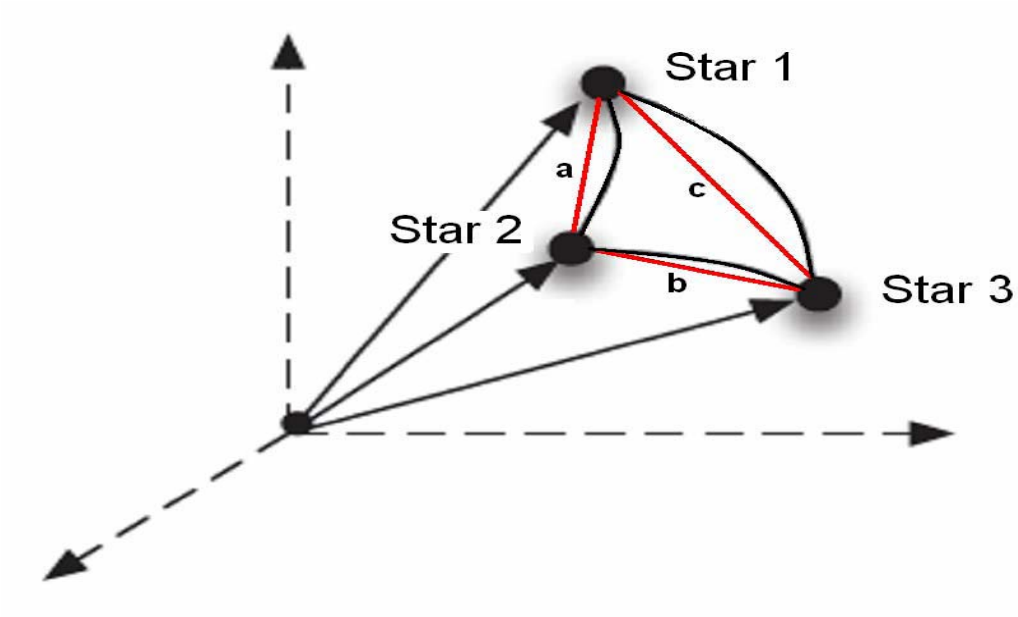

Figure 10. Spherical Triangles Method for Attitude Determination (From Diaz, 2006).

The three unit vectors to the stars within the FOV, allow the area of the spherical triangle to be calculated by:

$$A = 4\tan^{-1}\sqrt{\tan\left(\frac{s}{2}\right)\tan\left(\frac{s-a}{2}\right)\tan\left(\frac{s-b}{2}\right)\tan\left(\frac{s-c}{2}\right)} \tag{9}$$

where S is still represented by Equation 4, and a, b, and c are:

$$a = \cos^{-1}\left(\frac{b_1 \bullet b_2}{|b_1||b_2|}\right) \tag{10}$$

$$b = \cos^{-1}\left(\frac{b_2 \bullet b_3}{|b_2||b_3|}\right) \tag{11}$$

$$c = \cos^{-1}\left(\frac{b_3 \bullet b_1}{|b_3||b_1|}\right) \tag{12}$$

Again, the equations 9, 10, 11, and 12 are valid in the ECI frame as well as the star tracker frame (Cole & Crassidus, Fast Star Pattern Recognition Using Spherical Triangles, 2004).

The polar moment is also valuable information to be obtained from each observed triangle. Two similar triangles may have similar areas or polar moments, but it is extremely unlikely that two triangles will have exactly the same polar moments and areas. The acquisition of two unique pieces of information from each triangle makes the algorithm extremely resistant to false attitude determinations from the star tracker using an erroneous triangle.

The polar moment of a triangle is obtained by breaking the spherical triangle into smaller triangles. The area of each of these smaller triangles is then multiplied by the square of the arc distance from the centroid of each smaller triangle, to the centroid of the overall triangle (Cole & Crassidus 2004). The spherical triangle's polar moment is then obtained by summing the results of each smaller triangle:

$$I_p = \sum \theta^2 dA \tag{13}$$

where dA is the smaller triangle area and θ^2 is the arc distance. The polar moment of each spherical triangle is calculated via a recursive algorithm that breaks the triangle into smaller triangles successively until the depth of recursion is met.

2. Summary of Star Identification Algorithms

The algorithms described perform the basic function of identifying stars within the FOV of the star tracker. Those identified stars are then matched to inertial stars

within the onboard database. The next step is to identify the attitude of the spacecraft from the star information of which more computation is necessary.

With the angles, planar triangles, or spherical triangles calculated from the observed stars, the next step is to match the calculated angle with angles between stars as observed from the ECI frame. The satellite will maintain a database of these inertial angles. The angles maintained in the satellite database are only angles that will fit within the FOV of the star tracker. For example, a star tracker with an eight degree FOV, then the database will maintain only those angles of eight degrees or less (Cole & Crassidus, 2004). The algorithm will search the inertial database and match the body-frame angle. Once the match has been made, the corresponding inertial star vectors are known.

The next step is to use the vector information obtained by the previous algorithms to determine the spacecraft attitude. Recognition of the stars as seen by the star tracker is the first step towards determing the position of the satellite. The satellite attitude is determined by applying rotational kinematics to solve the attitude determination problem.

E. ROTATIONAL KINEMATICS

Kinematics is describing the orientation of a body that is in rotational motion. There are no forces involved in the spacecraft's rotation. With discussing the rotation of an object, it is useful to describe its orientation by an imaginary frame fixed in the body of the craft. The moving frame of the spacecraft must also be referenced in relation to a fixed Cartesian coordinate system. If this reference frame is a non-moving frame, then it is referred to as an *inertial* frame.

To accurately determine where the spacecraft is, reference frames are designated in the spacecraft and the Earth for clarification. The main frames used are the Earth-Centered Inertial (ECI) reference frame and the body-frame of the satellite. These frames will be used throughout for attitude determination.

The stationary frame of reference is the ECI frame with the XYZ frame positioned at the center of the Earth. Figure 11, taken from http://spaceflight.nasa.gov/realdata/elements/graphs.htm, illustrates the ECI frame. The X axis of the ECI frame points in the direction of the vernal equinox (Curtis, 2005).

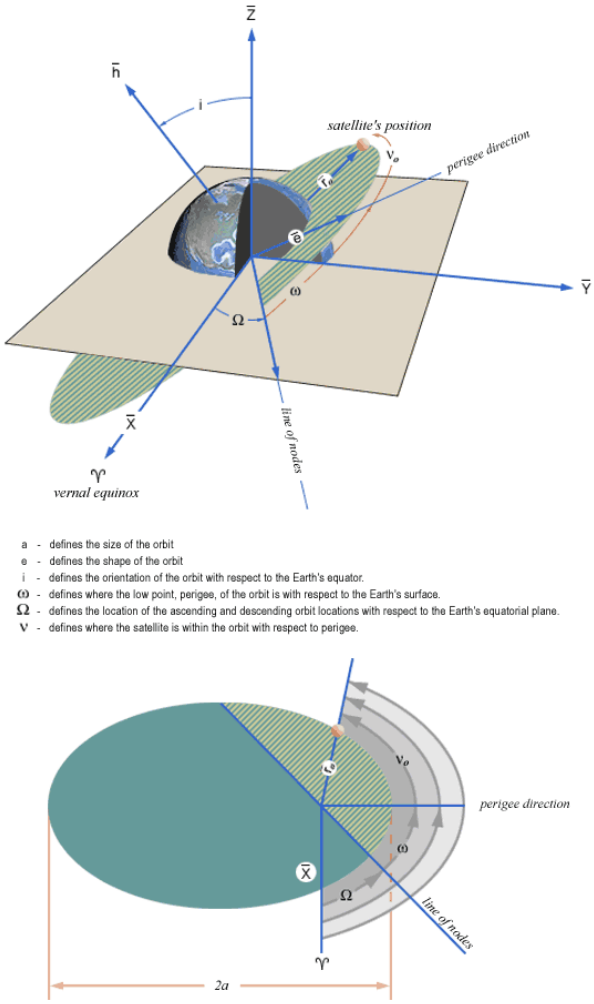

Figure 11. Earth-Centered Inertial (ECI) frame (From www.spaceflight.nasa.gov).

This frame will not move as the Earth rotates, therefore this fixed frame is used for determining star positions, all of which will be referenced from this coordinate system. The Earth is assumed to be a simple sphere in this case.

The body frame of the satellite is the frame associated with a satellite in orbit about the Earth. Figure 12 depicts the body frame of a vessel. The body frame is fixed

with respect to its vessel, and is used for attitude determination of the vessel. An image of a satellite in orbit with its respective body frame with respect to the ECI frame has been included as Figure 13 (Diaz, 2006).

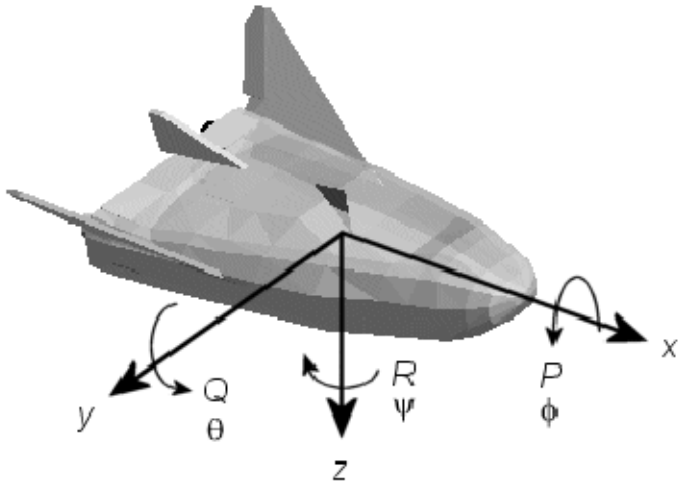

Figure 12. Satellite or spacecraft body frame (From www.mathworks.com).

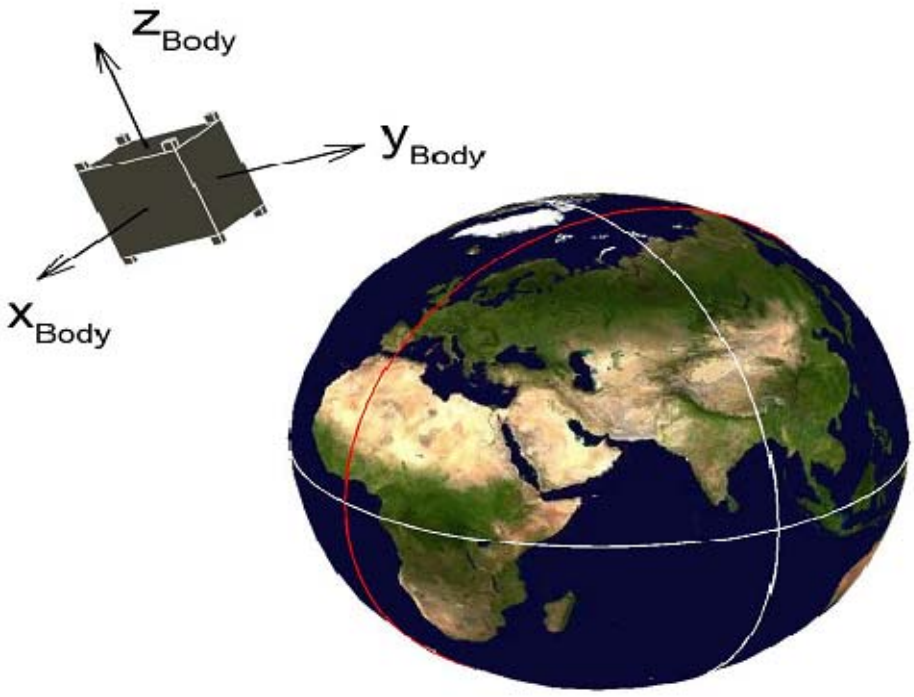

Figure 13. Body Frame with respect to the Earth Centered Inertial (ECI) Frame (From Diaz, 2006).

The ECI frame, as well as the body frame of the satellite, is fundamentally important to the attitude determination of the satellite. The star tracker provides the means to solve the relationship of the body frame of the satellite with respect to that of the ECI frame. However, the stars imaged by the star tracker must be matched to the satellite's onboard star catalog before any attitude solution is obtained.

1. Direction Cosine Matrix

In this case we have two different frames; one moving with respect to the other. The non-rotating geocentric equatorial frame provides an inertial frame for the two-body problem of a satellite in orbit. In this case, the ECI frame will be called the **B** frame while the satellite's body-frame is the **A** frame. The orthogonal **A** frame will have unit vectors $\begin{bmatrix}\vec{a}_1 & \vec{a}_2 & \vec{a}_3\end{bmatrix}$ while the B frame will have units vectors of $\begin{bmatrix}\vec{b}_1 & \vec{b}_2 & \vec{b}_3\end{bmatrix}$. The vectors of B are related to the frame A by the following:

$$\begin{aligned}\vec{b_1} &= C_{11}\vec{a_1} + C_{12}\vec{a_2} + C_{13}\vec{a_3}\\ \vec{b_2} &= C_{21}\vec{a_1} + C_{22}\vec{a_2} + C_{23}\vec{a_3}\\ \vec{b_3} &= C_{31}\vec{a_1} + C_{32}\vec{a_2} + C_{33}\vec{a_3}\end{aligned} \tag{14}$$

where $C_{ij} = \vec{b}_i \bullet \vec{a}_j$ is the cosine of the angle between vectors $\vec{b}_i$ and, $\vec{a}_i$ and is referred to as the *direction cosine matrix* (Wie, 1998). Equations 14 can be written in matrix format and are given as Equation 15:

$$\begin{bmatrix}\vec{b_1}\\ \vec{b_2}\\ \vec{b_3}\end{bmatrix} = \begin{bmatrix}C_{11} & C_{12} & C_{13}\\ C_{21} & C_{22} & C_{23}\\ C_{31} & C_{32} & C_{33}\end{bmatrix}\begin{bmatrix}\vec{a_1}\\ \vec{a_2}\\ \vec{a_3}\end{bmatrix} = C^{B/A}\begin{bmatrix}\vec{a_1}\\ \vec{a_2}\\ \vec{a_3}\end{bmatrix} \tag{15}$$

where $C^{B/A} = \begin{bmatrix}C_{ij}\end{bmatrix}$ is the aforementioned direction cosine matrix. The direction cosine matrix mathematically describes the orientation of B relative to A (Wie, 1998). The direction cosine matrix, $C^{B/A}$, will also be referred to as the direction cosine matrix (DCM) or the transformation matrix of B from A throughout this report. The orientation of frame B relative to A is written mathematically as:

$$C^{B/A} = \begin{bmatrix} \vec{b}_1 \bullet \vec{a}_1 & \vec{b}_1 \bullet \vec{a}_2 & \vec{b}_1 \bullet \vec{a}_3 \\ \vec{b}_2 \bullet \vec{a}_1 & \vec{b}_2 \bullet \vec{a}_2 & \vec{b}_2 \bullet \vec{a}_3 \\ \vec{b}_3 \bullet \vec{a}_1 & \vec{b}_3 \bullet \vec{a}_2 & \vec{b}_3 \bullet \vec{a}_3 \end{bmatrix} \equiv \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vec{b}_3 \end{bmatrix} \bullet \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & \vec{a}_3 \end{bmatrix} \tag{16}$$

2. Euler Angles

For defining or orienting a spacecraft in orbit, Euler's eigenaxis rotation theorem is used. Euler's eigenaxis theorem claims that a rigid body rotated about an axis fixed to the spacecraft body which is stationary in an inertial reference frame, the spacecraft attitude can be aligned from any given orientation to another orientation (Wie, 1998). This is commonly called body-axis rotation, which involves successive rotations about a body-fixed reference frame (Wie, 1998).

Body-axis rotation involves three consecutive rotations. The first rotation is about any axis, while the second rotation is about either of the two remaining body axes. The third and final rotation is about the last axis, which has not been rotated. The number of rotations means there are twelve sets of Euler angles for successive rotations about axes fixed on the spacecraft body. These rotations defining an orientation of frame B relative to frame A are described mathematically:

$$\begin{bmatrix} \vec{a}'_1 \\ \vec{a}'_2 \\ \vec{a}'_3 \end{bmatrix} = \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{a}_3 \end{bmatrix} = C_3(\theta_3) \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{a}_3 \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} \vec{a}''_1 \\ \vec{a}''_2 \\ \vec{a}''_3 \end{bmatrix} = \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \vec{a}'_1 \\ \vec{a}'_2 \\ \vec{a}'_3 \end{bmatrix} = C_2(\theta_2) \begin{bmatrix} \vec{a}''_1 \\ \vec{a}''_2 \\ \vec{a}''_3 \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vec{b}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} \vec{a}''_1 \\ \vec{a}''_2 \\ \vec{a}''_3 \end{bmatrix} = C_1(\theta_1) \begin{bmatrix} \vec{a}''_1 \\ \vec{a}''_2 \\ \vec{a}''_3 \end{bmatrix} \tag{19}$$

where A' and A" are two intermediate frames with unit vectors of $\left[\vec{a}_1' \quad \vec{a}_2' \quad \vec{a}'_3\right]$ and $\left[\vec{a}_1'' \quad \vec{a}_2'' \quad \vec{a}_3''\right]$ (Wie, 1998). The angles of these rotations $\left[\theta_1 \quad \theta_2 \quad \theta_3\right]$ are referred to as Euler Angles or the roll, pitch, and yaw angles (Wie, 1998). The direction cosine for this rotation from A to B is the product of the individual rotations as defined below as Equation 20:

$$C^{B/A} = \left[C_{ij}\right] \tag{20}$$

This direction cosine matrix, or attitude determination matrix, is key to understanding the orientation of a satellite body frame with respect to the inertial or ECI axis. It is easier to refer to this matrix as A for simplicity.

3. Euler's Eigenaxis Rotation

As mentioned previously, Euler's eigenaxis theorem states that rotation of a rigid body about an axis fixed to the body of the spacecraft and stationary in an inertial reference frame, the spacecraft's attitude can be shifted from any given orientation to another orientation (Wie, 1998). This axis of rotation orientation is the same in the inertial reference frame as the body frame. This axis is called the spacecraft's *Euler axis* or *eigenaxis* (Wie, 1998).

Using frames A and B in the previous section, the orientation of is described via the Euler axis. The orientation of B with respect to A is described by a unit vector $[\vec{e}]$ along the Euler axis with a rotation angle of θ (Wie, 1998). Mathematically this is described as follows:

$$\vec{e} = e_1\vec{a}_1 + e_2\vec{a}_2 + e_3\vec{a}_3 \tag{21}$$

$$= e_1\vec{b}_1 + e_2\vec{b}_2 + e_3\vec{b}_3 \tag{22}$$

where e_i are the direction cosines of the Euler axis for frames A and B.

4. Quaternions

Euler angles are used to define the orientation of a spacecraft, but quaternions are also used to define the attitude of a spacecraft. The equations relating Euler angles to quaternions are defined in Equation 23:

$$\begin{aligned} q_1 &= e_1 \sin\left[\frac{\theta}{2}\right] \\ q_2 &= e_2 \sin\left[\frac{\theta}{2}\right] \\ q_3 &= e_3 \sin\left[\frac{\theta}{2}\right] \\ q_4 &= \cos\left[\frac{\theta}{2}\right] \end{aligned} \tag{23}$$

or

$$\tilde{q} = \begin{Bmatrix} Q \\ q \end{Bmatrix} = \begin{Bmatrix} \tilde{e}\sin\left(\theta/2\right) \\ \cos\left(\theta/2\right) \end{Bmatrix}$$

where θ is again the rotation angle about the Euler axis (Wie, 1998). Since a four dimensional vector is used to describe the attitude of the spacecraft in three dimensions, the quarternion components are not independent of each other as can be seen in Equation 23 (Crassidis & Junkins, 2004). The quaternions are also constrained by the following relationship:

$$\begin{aligned} &q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \\ &\text{or} \\ &\tilde{q}^T \tilde{q} = 1 \end{aligned} \tag{24}$$

due to the fact that the following is true:

$$e_1^2 + e_2^2 + e_3^2 = 1 \tag{25}$$

The ability to link quaternions to a direction cosine matrix is a significant benefit for calculations. The use of quaternions eliminates use of trigonometric functions and the singularities that result. The quaternions are also ideal for calculation onboard a spacecraft since only products are calculated (Wie, 1998). Calculating the direction cosine matrix from quaternions is done by the equation:

$$C^{B/A} = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2+q_3q_4\right) & 2\left(q_1q_3+q_2q_4\right) \\ 2\left(q_2q_1+q_3q_4\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3+q_1q_4\right) \\ 2\left(q_3q_1+q_2q_4\right) & 2\left(q_3q_2+q_1q_4\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix} \tag{26}$$

or in Euler angles:

$$A = \begin{pmatrix} c\psi c\theta & s\psi c\phi + c\psi s\theta s\phi & s\psi c\phi - c\psi s\theta s\phi \\ -s\psi c\theta & c\psi c\phi - s\psi s\theta s\phi & c\psi s\phi + s\psi s\theta c\phi \\ s\theta & -c\theta s\phi & c\theta c\phi \end{pmatrix} \tag{27}$$

where $c\psi \equiv \cos\psi$, $s\phi \equiv \sin\phi$, etc, (Crassidis & Junkins, 2004).

5. Euler Angles and Quaternions in Attitude Determination

The body frame of the object in Figure 6 or the small boxlike satellite in Figure 7 can be mathematically mapped from the reference frame which is the Earth shown in Figure 5 to the body frame. Referring to Equation 2 and replacing arbitrary frames A and B with the ECI frame and the body frame of the satellite, the attitude of the satellite is given as a direction cosine matrix A. The body frame b has components $\tilde{b} = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T$ and the ECI frame will have its coordinates as $\tilde{r} = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$.

Equation 14 is used to map the frames of the satellite to the ECI frame. The inertial vectors $\boldsymbol{b_i}$ are then used to determine the attitude of the satellite by:

$$\tilde{b}_i = Ar_i \tag{28}$$

where A is the orthogonal and proper direction cosine matrix and r_i are the star vectors in the body frame (Cole & Crassidus, 2006).

Naturally, all angles calculated using equation (3) will have some error in them, so Equation 27 is not satisfactory. To account for errors, most of the error is concentrated on a small area about the direction of Ar_i, and therefore the sphere containing that point is approximated as a tangent plane, which is represented by the following equation (Cole & Crassidus, 2004):

$$\begin{aligned} &\tilde{b}_i = Ar_i + v_i, \\ &v_i^T Ar_i = 0 \end{aligned} \tag{29}$$

where $\tilde{b}_i$ is the *i*th measurement and the sensor error $\mathbf{v}_i$ is approximately Gaussian (Cole & Crassidus, 2004). Therefore all angle measurements will contain some error and this error must be accounted for.

F. ATTITUDE DETERMINATION FROM VECTOR OBSERVATIONS

The problem of attitude determination is obtaining the correct orthogonal rotation matrix, so that the measured observations in the sensor frame match the reference frame observations mapped by that matrix into the sensor frame (Crassidis & Junkins, 2004). The measured vectors are the aforementioned body-frame vectors to imaged stars while the reference vectors are those same stars referenced from the ECI frame.

The stars imaged in the FOV of the star tracker have now been paired to stars in the inertial frame by the star pattern recognition algorithms, but the attitude of the spacecraft is still unknown. For this section, the inertial reference unit vectors are represented by $\hat{V}_1 \ldots \hat{V}_n$, and the body frame unit vectors are represent by $\hat{W}_1 \ldots \hat{W}_n$ (Shuster & Oh, 1981). Therefore, an orthogonal matrix *A* is needed that satisfies:

$$A\hat{V}_i = \hat{W}_i, \qquad (i = 1, \ldots, n) \tag{30}$$

Due to measurement errors and corruption in both the star tracker measurements and errors in the inertial vectors, there is no exact solution for *A*. Therefore an approach is needed to select an *A* that matches $\hat{V}_i$ to $\hat{W}_i$. This is known as "Wahba's Problem."

Wahba's problem is the estimation of a satellite's attitude by using direction cosines (Wahba, 1966). Given two sets of points, in this case $\hat{V}_1 \ldots \hat{V}_n$ and $\hat{W}_1 \ldots \hat{W}_n$ where n ≥ 2, find a rotation matrix *A* which aligns the first set of vectors into the best least squares coincidence with the second set of vectors (Wahba, 1966). Mathematically, a matrix *A* minimizes:

$$\sum_{j=1}^{n} \left\| \hat{W}_j - A\hat{V}_j \right\|^2 \tag{31}$$

where $\|\bullet\|$ denotes the Euclidean norm. Equation 31 is also represented in the terms of a cost or loss function as:

$$L(A) = \frac{1}{2}\sum_{i=1}^{n} a_i \left| \hat{W}_i - A\hat{V}_i \right|^2 \tag{32}$$

subject to the constraint:

$$AA^T = I_{3x3} \tag{33}$$

The quadratic loss function in the attitude matrix can be transformed into a quadratic loss function in the corresponding quaternion (Shuster & Oh, 1981). Wahba presents a least-squares criterion to define the best estimate for an orthogonal matrix A that minimizes the cost function represented by Equation 32

G. ATTITUDE DETERMINATION ALGORITHMS

There are many different types of attitude determination algorithms for star trackers in use today, but a common type used is a class that estimates the four Euler symmetric parameters that form the quaternion in Equation 23 (Weiss, Bar-Itzhack, & Oshman, 2005). The quaternion outputs of these algorithms are extremely popular as it is the minimal non-singular set for global attitude description (Weiss, Bar-Itzhack, & Oshman, 2005). The quaternion also provides an attitude matrix, which is quadratic in the parameters and it also is free of transcendental trigonometric functions (Crassidis & Junkins, 2004). The optimal estimator of the quaternion can be used to solve the constrained least-squares Wahba problem identified in Equation 31 (Weiss, Bar-Itzhack, & Oshman, 2005).

Other algorithms used in solving Wahba's problem by obtaining the quaternion is the TRIAD algorithm as well as the Quaternion Estimator (QUEST) algorithm. The TRIAD and QUEST algorithms each provide quaternions as well as the direction cosine matrix of the satellite. The TRIAD algorithm is fairly simplistic, without requiring any inversion of matrices, while the QUEST algorithm requires fairly complex eigenvalue calculations.

1. Linear Least Squares Attitude Determination

a. Least Squares Problem Setup

Previously, Equation 29 estimated the position of the observed vectors to actual star vectors with some error v_i. However, the error or residual errors are assigned to each measurement of r_i. Therefore, Equation 29 becomes Equation 34

$$\begin{aligned} \tilde{b}_i &= A\tilde{r}_i + \tilde{e}_i \\ \tilde{b}_i &= \begin{bmatrix} \tilde{b}_1 & \tilde{b}_2 & \dots & \tilde{b}_n \end{bmatrix} \\ \tilde{e}_i &= \begin{bmatrix} \tilde{e}_1 & \tilde{e}_2 & \dots & \tilde{e}_m \end{bmatrix} \\ \tilde{r}_i &= \begin{bmatrix} \tilde{r}_1 & \tilde{r}_2 & \dots & \tilde{r}_m \end{bmatrix} \end{aligned} \tag{34}$$

where b_i is the measured values for the inertial star vectors and e_i are the residual errors for each star tracker measurement of r_i.

Using Gauss's principle of least squares, it is desired to obtain an A that minimizes the residual errors. Solving for the residual errors we obtain:

$$\tilde{e}_i = A\tilde{r}_i - \tilde{b}_i \tag{35}$$

Using Equation 34 as a cost function of residual errors is (Crassidis & Junkins, 2004):

$$J = \frac{1}{2} e^T e \tag{36}$$

Or, by substitution of equation 22 into equation 23 and dropping the subscripts for clarity:

$$J = \frac{1}{2}\left(\tilde{b}^T\tilde{b} - 2\tilde{b}^T A\tilde{r} + \tilde{r}^T A^T A\tilde{r}\right) \tag{37}$$

There are two requirements for minimizing globally the quadratic function: 1) a necessary condition and 2) a sufficient condition (Crassidis & Junkins, 2004). The necessary condition and sufficient conditions are defined as:

$$\nabla_r J = \begin{bmatrix} \dfrac{\partial J}{\partial \tilde{r}_1} \\ \vdots \\ \dfrac{\partial J}{\partial \tilde{r}_n} \end{bmatrix} = A^T A\tilde{r} - A^T\tilde{b} = 0 \tag{38}$$

$$\nabla_r^2 J \equiv \frac{\partial^2 J}{\partial r \partial r^T} = A^T A \tag{39}$$

where $A^T A$ must be positive definite (Crassidis & Junkins, 2004). In Equation 38, $\nabla_r J$ is the Jacobian and $\nabla_r^2 J$ is the Hessian in Equation 39. The matrix A is positive definite when it the matrix has a maximum rank (n) (Crassidis & Junkins, 2004). The quadratic function J is a performance surface in n + 1 dimensional space with a convex shape of an n-dimensional parabola with a single distinct minimum (Crassidis & Junkins, 2004).

From the necessary conditions defined in Equation 38 above, the "normal equations" are:

$$\left(A^T A\right)\tilde{r} = A^T \tilde{b} \tag{40}$$

If there are n independent observation equations, therefore the rank of A is n, making $A^T A$ positive definite (Crassidis & Junkins, 2004). With equation $A^T A$ positive definite, $\left(A^T A\right)$ is invertible and an explicit solution for the optimal solution is obtained. Therefore $\tilde{r}$ is solved by:

$$\tilde{r} = \left(A^T A\right)^{-1} A^T \tilde{b} \tag{41}$$

Equation 41 is the matrix equivalent of Gauss' original "equations of condition" in index/summation notation (Crassidis & Junkins, 2004).

Naturally, an inverse of $A^T A$ must exist to find a solution for $\tilde{r}$. The inverse exists only if there number of linearly independent observations is equal to or greater than the number of unknowns. In least squares, the order of the matrix inverse is equal to the number of unknowns, not the number of measurement observations (Crassidis & Junkins, 2004). An example of attitude determination with a star will illustrate these principles.

b. Least Squares Solution

Using Figure 6, the camera of the star tracker observes two stars within its FOV. The unit vectors of these stars in the star tracker reference frame are calculated with Equation 1 and are designated $\tilde{r}_1$ and $\tilde{r}_2$. These two stars have unit vectors in the

inertial frame as well, $\hat{R}_1$ and $\hat{R}_2$. For this example, we will only use one of the stars for calculations. The inertial coordinates of the star are matched the body coordinates by a direction cosine matrix *A*. Therefore the equation is

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{42}$$

Rearranging Equation 42, the equation becomes:

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{pmatrix} a_{11}r_x & a_{12}r_y & a_{13}r_z \\ a_{21}r_x & a_{22}r_y & a_{23}r_z \\ a_{31}r_x & a_{32}r_y & a_{33}r_z \end{pmatrix} \tag{43}$$

Equation 43 can be rearranged to take the following form:

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} r_x & r_y & r_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_x & r_y & r_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_x & r_y & r_z \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} \tag{44}$$

Equation 44 now takes the form of the normal matrix equation:

$$\hat{y} = A\hat{x} \tag{45}$$

with

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \hat{y}, \qquad \begin{bmatrix} r_x & r_y & r_z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_x & r_y & r_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_x & r_y & r_z \end{bmatrix} = A, \hat{x} = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix}$$

The $\hat{y}$ vector comprises the known inertial coordinates to the star, the A matrix is the known body frame vector to the same star, with the $\hat{x}$ comprising the elements of the direction cosine matrix being the only unknown quantity. Now Equation 45 can be solved by inserting the elements of the matrix equation into Equation 46 to form the least squares problem:

$$\hat{x} = \left(A^T A\right)^{-1} A^T \hat{y} \tag{46}$$

The vector $\hat{x}$ of the direction cosine matrix is simply reshaped into the usual form to get the direction cosine matrix of the frames into its usual form as in Equation 41.

2. TRIAD Algorithm for Attitude Determination

The TRIAD algorithm is a deterministic solution that generates a direction cosine matrix between two coordinate systems when two vectors are given in each of the particular coordinate systems (Bar-Itzhack & Harman). Applying this algorithm to the attitude determination problem is fairly straightforward. The star tracker needs only to see two stars within its FOV to determine two unit vectors using Equation 1. These are referred to as the observed vectors (Shuster & Oh, 1981). The other two unit vectors, or reference vectors, are found using the angle, planar triangles, or spherical triangles algorithms defined previously.

Using the TRIAD algorithm, two non-parallel unit vectors to stars in the inertial frame as well as two non-parallel unit vectors in the star tracker frame are obtained. Using the same designation when describing Wahba's problem in Equation 30, these vectors are identified as $\hat{V}_1$ and $\hat{V}_2$ for inertial stars with two body frame vectors from the star tracker as $\hat{W}_1$ and $\hat{W}_2$. The algorithm then finds an orthogonal matrix A, which becomes the attitude matrix for the satellite finds the orientation difference between the two systems (Shuster & Oh, 1981). The equations that the algorithm must satisfy are:

$$A\hat{V}_1 = \hat{W}_1 \qquad A\hat{V}_2 = \hat{W}_2 \tag{47}$$

The algorithm then requires computation of the following column matrices or triads (Shuster & Oh, 1981):

$$\begin{aligned} \hat{r}_1 &= \hat{V}_1 \qquad \hat{r}_2 = \left(\hat{V}_1 \times \hat{V}_2\right) \Big/ \left|\hat{V}_1 \times \hat{V}_2\right| \\ \hat{r}_3 &= \left(\hat{V}_1 \times \left(\hat{V}_1 \times \hat{V}_2\right)\right) \Big/ \left|\hat{V}_1 \times \hat{V}_2\right| \end{aligned} \tag{48}$$

$$\begin{aligned} \hat{s}_1 &= \hat{V}_1 \qquad \hat{s}_2 = \left(\hat{W}_1 \times \hat{W}_2\right) \Big/ \left|\hat{W}_1 \times \hat{W}_2\right| \\ \hat{s}_3 &= \left(\hat{W}_1 \times \left(\hat{W}_1 \times \hat{W}_2\right)\right) \Big/ \left|\hat{W}_1 \times \hat{W}_2\right| \end{aligned} \tag{49}$$

With Equations 48 and 49 defined, there exists a unique orthogonal matrix that satisfies:

$$A\hat{r}_i = \hat{s}_i \quad \left(i = 1,2,3\right) \tag{50}$$

which is defined as:

$$A = \sum_{i=1}^{3} \hat{s}_i \hat{r}_i^T \tag{51}$$

The triads are then constructed into matrices for further computation. A reference matrix is made consisting of the reference triads while an observed matrix is likewise constructed of observed triads. The matrices are:

$$M_{ref} = \left[\hat{r}_1 \vdots \hat{r}_2 \vdots \hat{r}_3\right] \quad M_{obs} = \left[\hat{s}_1 \vdots \hat{s}_2 \vdots \hat{s}_3\right] \tag{52}$$

where M_{ref} and M_{obs} matrices are 3x3 matrices. The attitude determination matrix is obtained by:

$$\begin{aligned} A &= M_{obs} M_{ref}^T \quad \text{or} \\ A &= r_1 \bullet s_1^T + r_2 \bullet s_2^T + r_3 \bullet s_3^T \end{aligned} \tag{53}$$

There are problems with the TRIAD algorithm though. The first vector has more prominence in determination of A. Some of the information in the second vector is discarded (Shuster & Oh, 1981). It is therefore necessary and best practice to obtain use the most accurate instrument to find the first vector of each set, in this case $\hat{V}_1$ and $\hat{W}_1$. Therefore, the first anchor (anchor vector) may be obtained by the star tracker, while the second vector cold come from the magnetometer (Bar-Itzhack & Harman).

3. QUEST Algorithm for Attitude Determination

The QUEST algorithm developed for the Magsat mission by Shuster is another method to solve Equation 32. The quadratic loss in the attitude matrix function of Equation 32 can be converted to a corresponding quaternion (Shuster & Oh, 1981). The result is that an eigenvalue equation is obtained that provides the quaternion (Shuster & Oh, 1981). This result is that the optimal quaternion is computed by a fast deterministic algorithm.

Equation 32, the loss function, is minimized when an optimal matrix A_{opt} is determined, however, we can also maximize a gain, *g*, which also solves the same equation. In Equation 32, the nonnegative a_i, $i = 1, \ldots, n$ are a set of weights (Shuster & Oh, 1981). Since the loss function may be scaled without affecting the resultant, A_{opt}, it is therefore possible to set:

$$\sum_{i=1}^{n} a_i = 1 \tag{54}$$

The corresponding gain function $g(A)$ is given as

$$g(A) = 1 - L(A) = \sum_{i=1}^{n} a_i \hat{W}_i^T A \hat{V}_i = q^T K q \tag{55}$$

It is easy to see that the loss, $L(A)$, function will be at a minimum when the gain function , $g(A)$ is at its maximum (Shuster & Oh, 1981). Equation 55 can be interpreted in the following way as well:

$$g(A) = \sum_{i=1}^{n} a_i tr\left[\hat{W}_i^T A \hat{V}_i\right] \tag{56}$$

where *tr* represents the trace operation performed in MATLAB. The matrix A is usually represented as quaternions since they are simpler to use.

To continue with this algorithm, several other quantities will need to be calculated to form the matrix *K* of Equation 55. The matrix is a 4x4 matrix that takes the following form:

$$K = \begin{pmatrix} S - \sigma I & Z \\ Z^T & \sigma \end{pmatrix} \tag{57}$$

where Z is a 3x1 vector, $S - \sigma I$ is a 3x3 matrix, Z^T is a 1x3 matrix, and σ is a scalar (Shuster & Oh, 1981). The matrix S is defined from the equation:

$$S = B + B^T = \sum_{i=1}^{n} a_i (\hat{W}_i \hat{V}_i^T + \hat{V}_i W_i^T)$$

where (58)

$$B = \sum_{i=1}^{n} a_i \hat{W}_i \hat{V}_i^T$$

The vector, Z, is defined as:

$$Z = \sum_{i=1}^{n} a_i \left(\hat{W}_i \times \hat{V}_i\right) \tag{59}$$

The quantity σ is the trB or:

$$\sigma = \sum_{i=1}^{n} a_i \hat{W}_i \bullet \hat{V}_i \tag{60}$$

Using these quantities, the gain function can be written in the following form by inserting them into Equation 55:

$$g(\tilde{q}) = \left(q^2 - Q \bullet Q\right) trB^T + 2\text{tr}\left[QQ^T B^T\right] + 2q\text{tr}\left[\underline{Q}B^T\right] \tag{61}$$

where

$$\underline{Q} = \begin{pmatrix} 0 & Q_3 & -Q_2 \\ -Q_3 & 0 & Q_1 \\ Q_2 & -Q_1 & 0 \end{pmatrix} \tag{62}$$

Using the matrix K and Equation 60, this produces a bilinear equation of the form:

$$g\left(\tilde{q}\right) = \tilde{q}^T K \tilde{q} \tag{63}$$

Using the constraint of Equation 24, the quaternion that maximizes Equation 62 can be used by implementing Lagrange multipliers (Shuster & Oh, 1981). A new gain function is defined. Using the notation of introduced by Shuster and Oh, this gain function is denoted as $g'\left(\tilde{q}\right)$. The gain function is written as:

$$g'\left(\tilde{q}\right) = \tilde{q}^T K \tilde{q} - \lambda \tilde{q}^T \tilde{q} \tag{64}$$

which is maximized without constraint (Shuster & Oh, 1981). The variable, λ, is used to satisfy this constraint. The verification is satisfied by differentiating which produces the equation:

$$K\tilde{q} = \lambda \tilde{q} \tag{65}$$

Therefore, the optimal quaternion is an eigenvector of the matrix K, and λ is an eigenvalue. The maximizing of $g'\left(\tilde{q}\right)$ will occur by choosing the eigenvector that corresponds to the largest eigenvalue of the matrix K (Shuster & Oh, 1981). Therefore in reality, Equation 65 takes the form of:

$$K\tilde{q}_{opt} = \lambda \tilde{q}_{opt} \tag{66}$$

4. Attitude Determination Algorithm Summary

The Least Squares, QUEST, and TRIAD algorithms have been introduced in the previous section with each having specific beneficial characteristics. The function $(A^T A)^{-1}$ is the core component of any least squares algorithm, but this calculation is very expensive (Crassidis & Junkins, 2004). It is possible to decompose the A matrix by QR decomposition or singular value decomposition. The singular value decomposition

algorithm is one of the most capable algorithms to compute least squares, yet the algorithm may be more expensive than the method introduced here. Due to cost of the Least Squares approach, it may be more beneficial to purse the TRIAD or QUEST algorithm. The QUEST and TRIAD algorithms have the ability to provide three-axis attitude determinations, without costly computations, which make them ideal for onboard attitude determination (Shuster & Oh, 1981). Yet, for moderate accuracy missions the more accurate QUEST algorithm may not be necessary since the QUEST algorithm has approximately twice the computation cost as the TRIAD algorithm (Shuster & Oh, 1981).

THIS PAGE INTENTIONALLY LEFT BLANK

III. SIMULATIONS

A. EXPERIMENT SETUP

Prior to performing any experimentation on hardware, the aforementioned star tracker star pattern recognition algorithms and attitude determination algorithms were tested using extensive simulations. All simulation runs were performed using MATLAB software in combination with images taken from a simulated star field. The star fields represent a scene from the inertial frame of reference and a view representing what a satellite in orbit would view.

1. Star Field Simulation

Using the Hipparcos catalog, three hundred of the brightest stars were identified and made into a two-dimensional star field. Using the right ascension and declination of each star, the right ascension and declination were transferred into x and y coordinates and plotted. The intensities for the stars were also varied according to their intensity listed in the catalog. The 300 stars were then imaged by a camera to provide input to the star pattern recognition algorithms. Figure 14 illustrates the star field designed for testing of the star pattern recognition algorithms. Naturally, 300 stars are only a fraction of the stars in the Hipparcos catalog as shown in Figure 3; however, this amount will be sufficient for the experiment.

The simulated star field provides an excellent image for testing star pattern recognition algorithms. The intensity of each star is at a wide range to provide varying brightness to the algorithms. The stars a dispersed to provide ample room between stars for ease of star recognition, yet some are extremely close which means the star pattern algorithm must be able to handle stars literally on top of each other.

Figure 14. Star Field.

B. STAR TRACKER SIMULATION SETUP

The first phase of the simulations involved setting up the equipment in the lab to support the star pattern recognition and attitude determination algorithms. A non-moving three-axis simulator was used to perform as a spacecraft. A camera was then installed to perform as a star tracker. Simulations were then completed to test the algorithms prior to installation of any new software onto the Three-Axis Spacecraft Simulator (TASS).

1. Simulated Star Tracker

A WAT-902H2 SUPREME (EIA) camera from Watec with an AF NIKKOR 50 mm lens from Nikon was used to act as a star tracker for the simulations. MATLAB code was developed to perform centroiding calculations on stars imaged within the camera's FOV. Using Equation 1, the unit vectors in the camera frame were calculated. Each pixel has a horizontal length of 8.4 μm and a vertical height of 9.8 μm. The focal

length, f, of the camera is 50 mm. These dimensions will be used in Equation 1 for calculating unit vectors to each star. Naturally, only a small portion of the star field was seen by the camera/star tracker. The camera with a narrow view lens installed is shown in Figure 15.

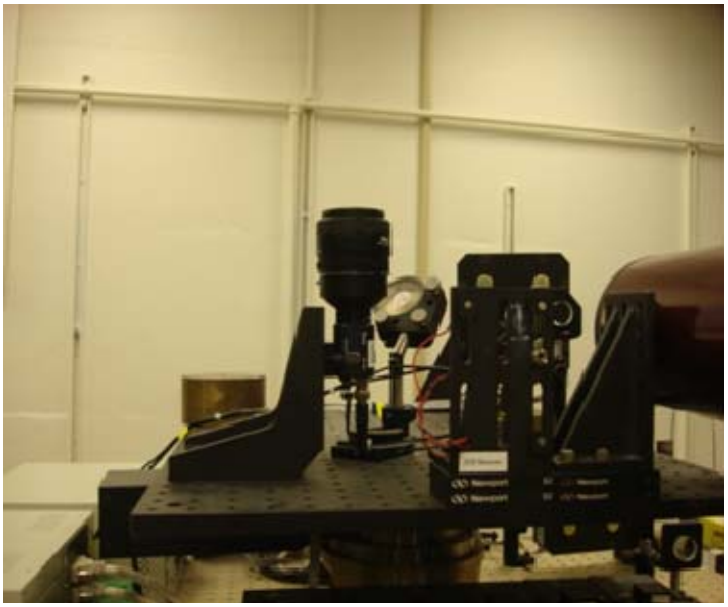

Figure 15. WAT-902H2 SUPREME (EIA) camera used as a star tracker.

2. Simulated Star Field

The star field in Figure 14 was imaged on a monitor installed on the ceiling of the lab. The monitor is seen directly above the TAS-2 as shown in Figure 27. The monitor is directly above the platform with the distance separating the camera and the monitor at 1.25 meters. By dimming the lights and installing curtains, the star field is easily visible to the camera. The camera in Figure 15 then captured the image for the experiments.

3. Star Field Detection Code

With the simulated star field in Figure 14 and the equipment in Figures 15 and 22, the star tracker imaged the stars using a code developed in MATLAB called monograb. The code has settings for the size of boxes used for stars detected and the intensity of star detected. The code parses through the image detecting the brightest pixels first. At each run through the image, a box is drawn around the detected bright pixel, then that box is

blacked out. The next run commences with the next brightest pixel is detected and a box of a designated area is drawn about that pixel. The next box is blacked out as well. The code parses through the image until all the brightest stars above a defined threshold are only blacked out boxes. The x and y coordinates or centroids, and brightness for each of these boxes, or stars, are stored by the code in arrays. The Matlab code provides the start for all future algorithms.

Figure 16 is an image of a star field with seven stars picked up by the centroiding algorithm. With the box size set at eight, seven of the eight stars are picked up. The eighth star is too close to the frame edge; therefore, a box cannot be drawn. The result is that only seven of the eight stars can be used for future algorithms even though the intensity of the star can be picked up by the code.

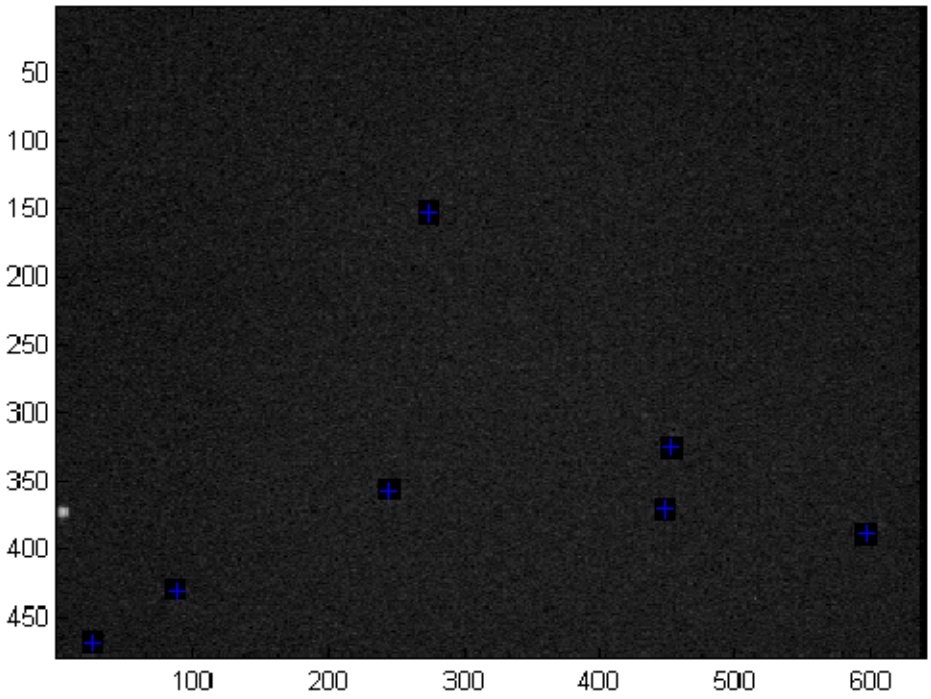

Figure 16. Star centroid plots using Matlab.

C. STAR FIELD ALGORITHM TESTING

1. Satellite Star Database

Prior to any attitude determination testing, the satellite must possess an onboard database of these stars as referenced to the inertial frame. The database must possess the attributes necessary for angle, planar triangles, or spherical triangles in storage. The database must also possess the unit vectors to each star $\begin{bmatrix} \hat{r}_1 & \hat{r}_2 & \dots & \hat{r}_n \end{bmatrix}$ associated with the angles, planar triangles, or spherical triangles. The variance of the angles, planar triangles, and spherical triangles must also be stored in the database as well.

a. Angle Database

Using the star field image shown in Figure 14, a database of angles was created. It is assumed for computer simulations that the database created with the zero attitude of the spacecraft simulator is in inertial frame. The origin of the inertial frame is not important as long as is the star objects are located at distant locations. In the laboratory environment with the stars displayed on the LCD screen in close proximity, the origin of the inertial frame must be located at the center of rotation of the spacecraft in order to serve as an inertial reference database. This fact is considered in actual experiments in Chapter IV. Figure 17 is a snapshot image of the database stars with the detected stars showing up as crosses for clarity. These stars will perform as the inertial stars for all computer simulations.

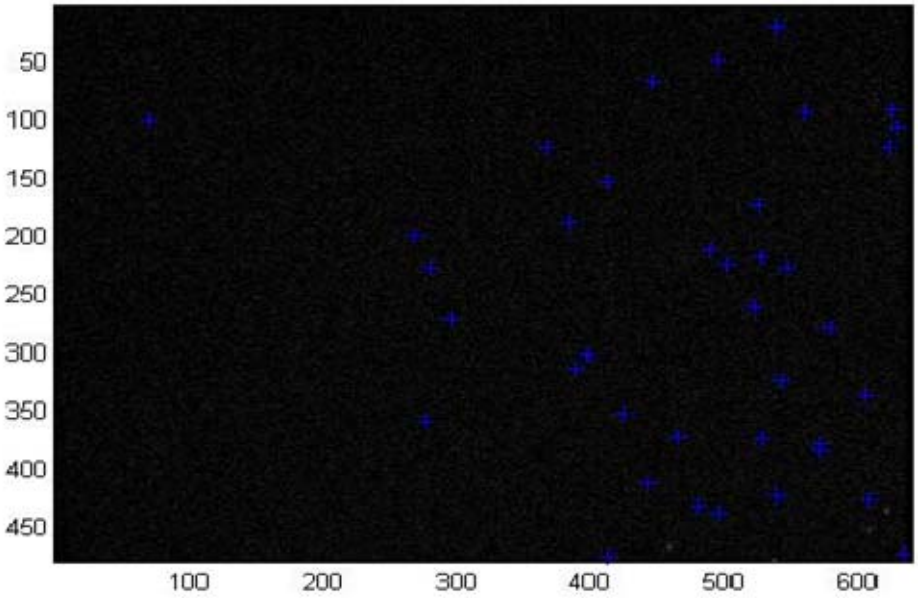

Figure 17. Database star field image.

The Matlab software processed the database image for development to run simulations and test the star pattern recognition algorithms developed in Chapter II. The first image taken represents an inertial database image of stars with subsequent images representing images taken by the satellite for attitude determination.

The database is formed by selecting a master star and performing angle calculations between the master star and all other stars in the image. Figure 18 is the mathematical representation of the image in Figure 17. The master star is the star with the highest intensity and is used as the reference star for each image when conducting calculations. The x and y coordinates and intensity for each star in the image is collected and the unit vector to each star is calculated. The master star is shown in Figure 18 as the star in blue. The angle between the master star and every other star is calculated. The angles between all other stars are calculated, as well leading to a database of over seven hundred angles. The variances of each angle are calculated, as well and stored for accuracy measurements later.

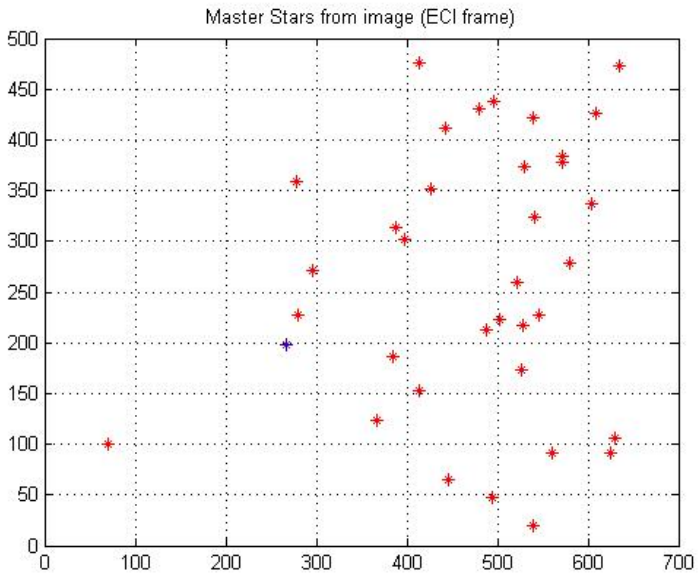

Figure 18. Database star field image by MATLAB.

The problem with the angle database is that there will be numerous redundant angles in it. For example, the angle between an arbitrary star one and two will also be the same as the one between two and one.

b. Planar Triangle Database

The next database developed was a planar triangle database. Again the image in Figure 17 was used to develop a MATLAB plot of the stars similar to Figure 18 with a master star. Using Equations 3, 4, 5, 6, and 7, a planar triangle database was created of all sets of planar triangles that could be developed from the visible stars. The polar moment and area of each triangle with their associated unit vectors are stored for each triangle.

The planar triangle database is far larger than an angle database due to more record keeping. Each triangle now has three stars and their respective unit vectors associated instead of two stars for each angle. Also, instead of one piece of data (the angle), there are now two pieces associated for each entry, area and polar moment. The results of the added complexity to the database is that searches within the database require far more time as well as more complex code.

c. Spherical Triangle Database

Using Figure 17, another image similar to Figure 18 was shot to begin calculations for a spherical triangle database. Equations 8, 9, 10, 11, 12, and 13 were used to develop entries for each triangles area and polar moment. As in the case of the planar triangles, the spherical triangles database must have far more entries than a simple angle database.

2. Star Pattern Recognition Algorithm Tests

a. Angle Simulations

After establishing all the databases, the next step was to create another image that would represent an image taken from the star tracker while the satellite is in orbit. Figure 20 represents the star field as seen from the camera or star tracker. The stars picked up by the star tracker camera are represented by blue crosses for clarity. The image shows five other stars on the right side and lower right that were too faint to be used for calculations.

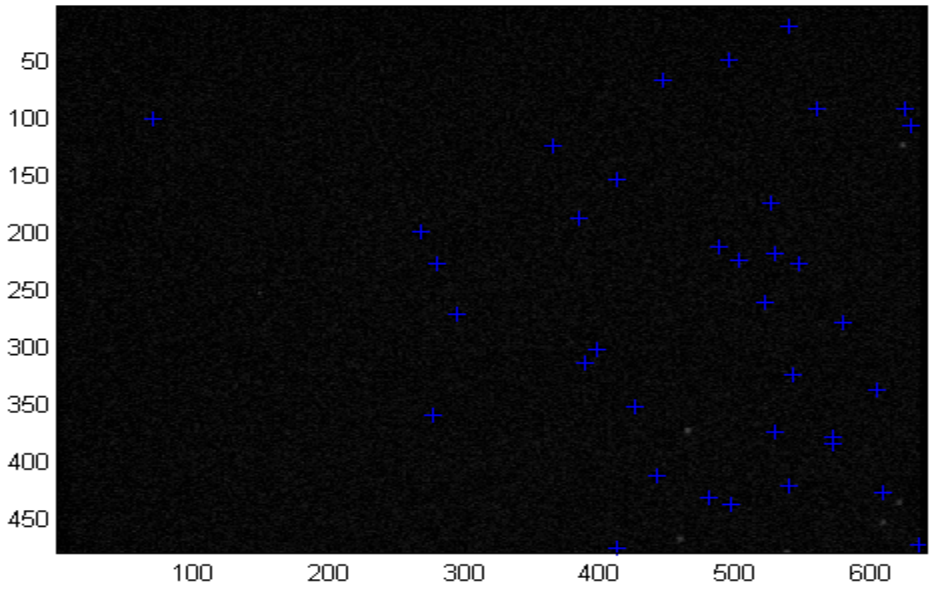

Figure 19. Star field as seen from the star tracker.

The star tracker image of Figure 19 is very close to the inertial image in Figure 17. This similarity is specifically to validate the algorithms used. Angles, planar triangles, and spherical triangles should be extremely close to the corresponding angles, planar triangles, and spherical triangles stored in the database. Therefore, every object should be able to match up to a corresponding object in the database. The result of attitude determination from these shots should result in a direction cosine matrix that is very close to the identity matrix or:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

MATLAB was again used to store the positions of each star in the star tracker image and perform the necessary calculations. Figure 20 is the representation of Figure 20 in MATLAB. The picked-up stars are shown in red, while the master star for the image is shown in blue. The star intensities in this image are somewhat different since the master star is not the same as the master star shown in Figure 19.

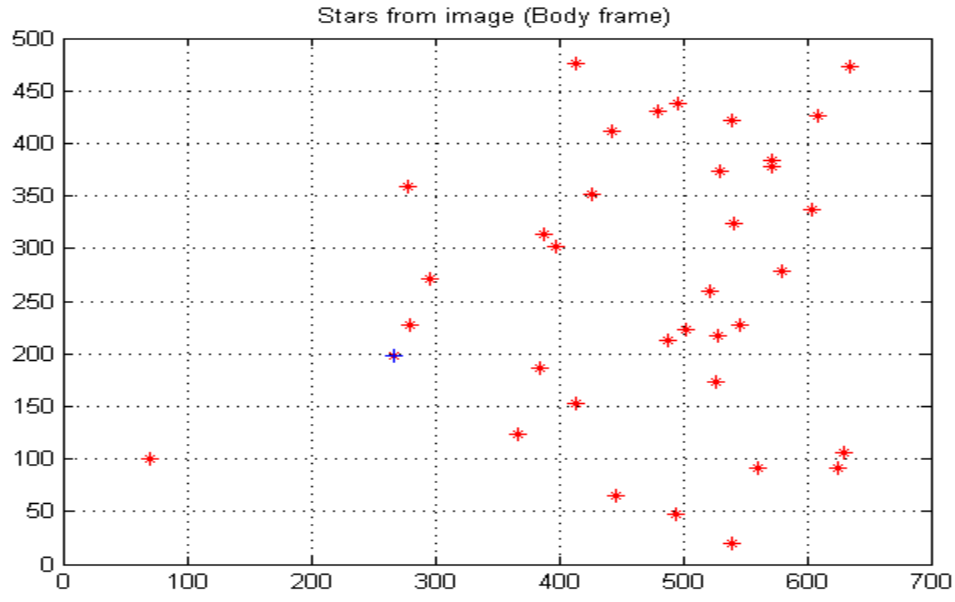

Figure 20. Star Tracker image in MATLAB.

The centroiding algorithm is again used to obtain all the necessary data from Figure 20. The master star in blue on Figure 20 is again used to calculate all the angles between the master star and the other stars in the image. For Figures 18 and 19, all the angles between all the stars and every other star are calculated. In the case of Figure 19 and 20, the only angles calculated are those between the master star and all the other stars in the image. The reason for this is that a star tracker will not be able to calculate several hundred angles and search through its onboard database without impacting the frequency of its solutions. Therefore, in this case it is only necessary to calculate angles between the master star and the other stars to ease the star matching search. When calculating the angles from an image, the master star is the brightest star and designated as star one in the algorithm. The star tracker frame angles are stored in an array along with the unit vectors to the stars in the camera frame.

The angle algorithm then compares the angles calculated by the star tracker to those angles stored in the onboard database. The variance in the inertial database is used for the accuracy in comparing the inertial angles to the camera angles. If a camera angle falls within the inertial angle +/- some accuracy, there is a match of the stars seen by the camera to the inertial stars.

Even though an angle is matched, there is some ambiguity that must be dealt with. For instance, which of the two body frame stars are associated with the two matched inertial stars? Also, if an angle is matched in the database, it must also have another match since two angles can be produced between two stars. For example if angle between the master star and a second star in the body frame is found, there also must another angle between the second star and the master star of exactly the same value.

There are possibilities of errors, since there are numerous angles that are very close to each other. Therefore some stars may be picked up erroneously and throw off the attitude determination of the satellite. When mapping the matched angles and their respective stars, not all the angles have matched up and a few angles are mismatched. By plotting all the picked inertial stars with their star tracker stars we can see those stars that are matched and those that are not. The inertial stars are in blue while the body frame stars are in red. As seen in Figure 21 there are four mismatched mis-

matched inertial and three body frame stars. The TRIAD, QUEST, or least-squares algorithms will attempt to match all stars, even the error stars, to form a best fit solution. These error stars must be filtered out to preclude an erroneous attitude solution. However, when searching a database of over seven hundred angles, these errors are very small when compared to the number of stars correctly matched.

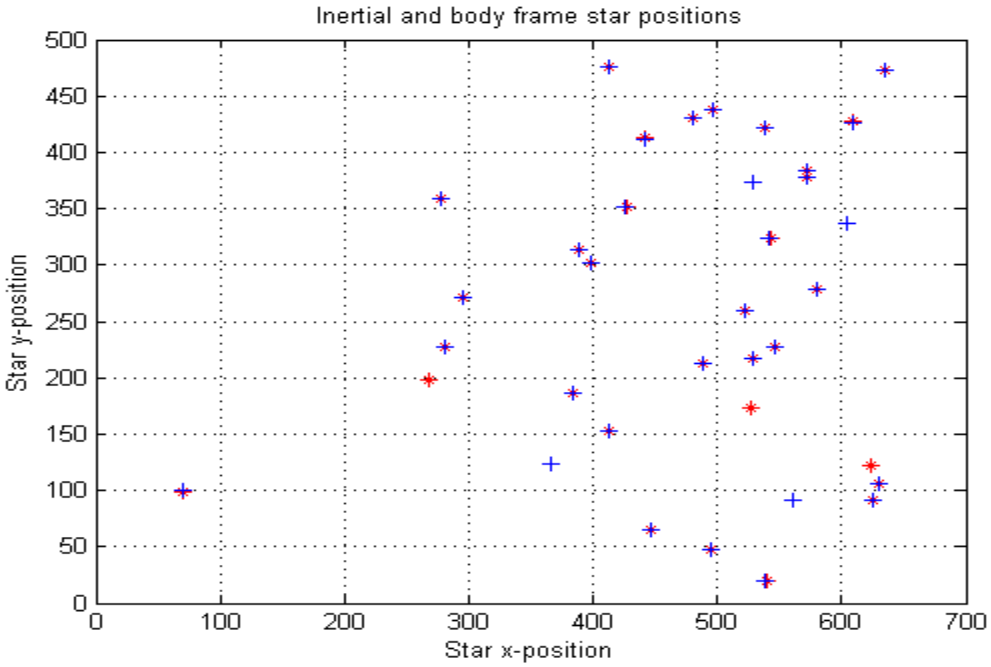

Figure 21. Inertial stars with their matches from the star tracker.

To prevent these error angles being included in the attitude calculations, filtering logic is developed to remove any erroneous angles and their stars. Whenever the angle algorithm is performed, a match of angles will always have an inertial star from the inertial angle matched to a similar star in the body frame. Since the master star in the body frame is designated as star one in the algorithm the inertial star appearing the most time in the angle matches is the star that matches the master star or star one in the image. The matched inertial star-to-star one is called the prime star for clarity. Therefore, it is easy now to match up the other remaining non number one stars in the body angle to the non-prime stars in the inertial angles.

After running the angle algorithm on the stars in Figure 21, the angle matches were obtained. A total of 26 angles matched. For each angle, there are two inertial stars and two body frame stars, one of which is the prime or master for the body frame stars. Each column in Table 1 represents an angle match. The top two rows are for the inertial stars, which are matched up to the body frame stars in the next two rows.

One may be tempted to continue right into attitude determination at this point, however, an incorrect solution would be acquired. Even though all stars are identified, the stars in the inertial frame must be correctly matched to the stars in the body frame of the satellite. Therefore, the results in Table 2 must be filtered to ensure all stars are matched correctly, and invalid angles with their stars are discarded.

Matched angles	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Inertial Stars	14	3	12	14	21	24	28	30	35	26	26	30	38	30	5	22	4	2	20	10	2	2	2	1	2	2
Inertial Stars	1	2	2	2	2	2	2	4	5	10	20	22	30	38	35	30	30	28	26	26	24	21	14	14	12	3
Body Frame Stars	21	4	9	13	20	27	22	33	16	6	13	36	19	19	16	36	33	22	13	6	27	20	13	21	9	4
Body Frame Stars	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 1. Matched body frame and inertial stars.

As shown in Table 1, the bottom row is all ones, representing the master star in the image. Therefore, the master star must correspond to the inertial star appearing the most in the angle matches. Table 2 illustrates the frequency of the inertial stars showing up in the angle matches. This table will allow us to determine which star in the inertial frame matches up to the body frame master star.

As seen in Table 2, the inertial star number showing up the most is star number two. Therefore Body Frame star one is Inertial Star two. Of the 38 stars imaged, the inertial star two appears twelve times or 23% of the time. With this information, we can match up all inertial star angles that have the number two with corresponding body with the correct body frame star. Therefore all angles that do not have body frame master star one matching up to an inertial angle with inertial star two in can be discarded as erroneous angles.

Inertial Star Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number times appearing	2	12	2	2	2	0	0	0	0	2	0	2	0	4	0
Percentage	3.8	23	3.8	3.8	3.8	0	0	0	0	3.8	0	3.8	0	7.7	0

Inertial Star Number	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Number times appearing	0	0	0	0	2	2	2	0	2	0	4	0	2	0	6
Percentage	0	0	0	0	3.8	3.8	3.8	0	3.8	0	7.7	0	3.8	0	11.5

Inertial Star Number	31	32	33	34	35	36	37	38
Number Times Appearing	0	0	0	0	2	0	0	2
Percentage	0	0	0	0	3.8	0	0	3.8

Table 2. Inertial star frequency.

By using the information gained in Table 2, using the body frame master star, we now can successfully match all the other body frame stars to their corresponding inertial stars. Keeping the accurate matches, all other matches where body frame star one does not match up to inertial star two are discarded as erroneous angles. Table 3 shows the remaining accurate matches. Focusing on angle three in Table 1, body frame star one is inertial star two, therefore body frame star four must equal inertial star three. This is repeated until all stars are identified correctly.

Table 3 is a listing of all stars correctly matched. Now, there are only seven star pairs that are accurate matches out of the original 38 stars in the image. These are the final correct angles and matches that now can be used for attitude determination.

Inertial Stars	2	3	12	14	21	24	28
Body Stars	1	4	9	13	20	22	27

Table 3. Correct body frame and inertial star matchups.

After filtering, the stars can be processed using one of the attitude determination algorithms. The filtered image of Figure 21 is displayed in Figure 22. Now all the imaged stars are matched up to an inertial star, as well as their corresponding unit vectors.

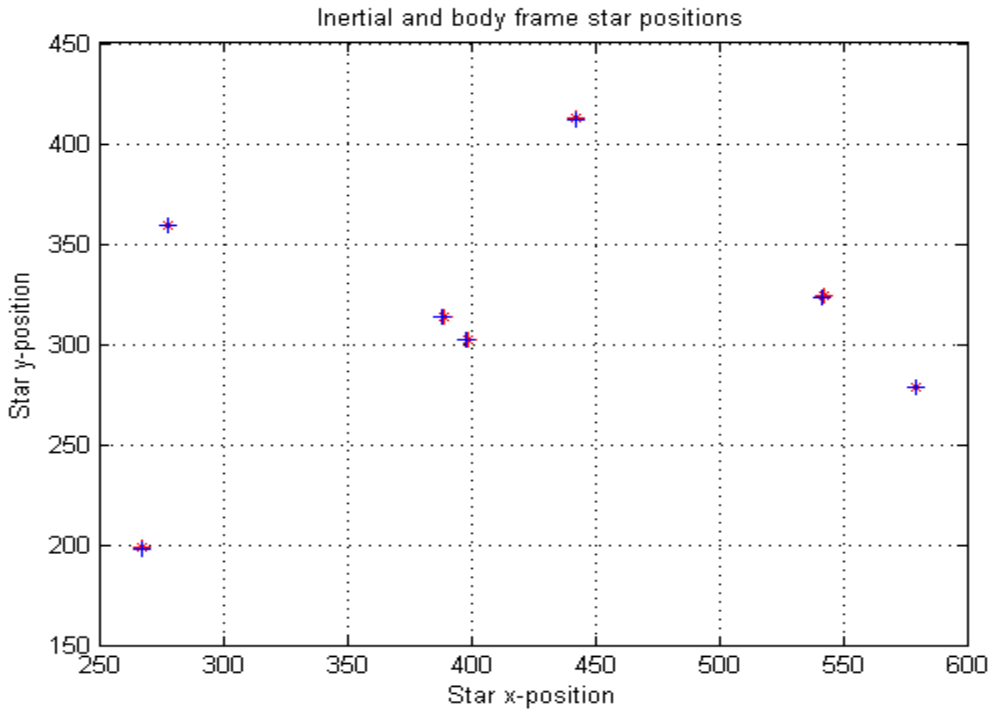

Figure 22. Filtered inertial stars with their matches from the star tracker

The results of the filtering are disappointing. The majority of the matchups are discarded resulting in a significant waste of processing. The simplicity of the algorithm is a benefit, yet with the majority of the results being discarded, the advantages of this algorithm may be few. The next step is to perform an attitude determination algorithm on the inertial and body vectors.

b. Planar Triangle Simulations

The next simulation test involved using planar triangles for star pattern identification. Figure 17, was used again as the database image to create a large array of areas, polar moments, and the inertial unit vectors associated with each triangle. The image in Figure 20 was used to simulate the stars viewed from an onboard star tracker. Equations 3, 4, 5, 6, and 7 were used to calculate all the body frame planar triangles.

The benefit of the planar triangles algorithm is that there is no ambiguity on matching up inertial to body triangles. Both the planar triangle area and the polar moment must match in order for the three stars of the inertial frame to be correlated to the

three stars in the image planar triangle. However, there are still possibilities of errors, therefore some filtering will be required to ensure error stars are removed.

Applying the planar triangle algorithm and a defined accuracy, all the body frame stars are matched up to the corresponding inertial stars. Figure 23 is the result of the inertial and body frame star matching. The inertial stars are represented by the red circles while the body frame stars matched are the blue crosses. Even though there are twenty-four exact matches, there are four inertial stars and nine body frame star without matches. From the indications of errors in Figure 23, the results of the planar triangle algorithm require filtering.

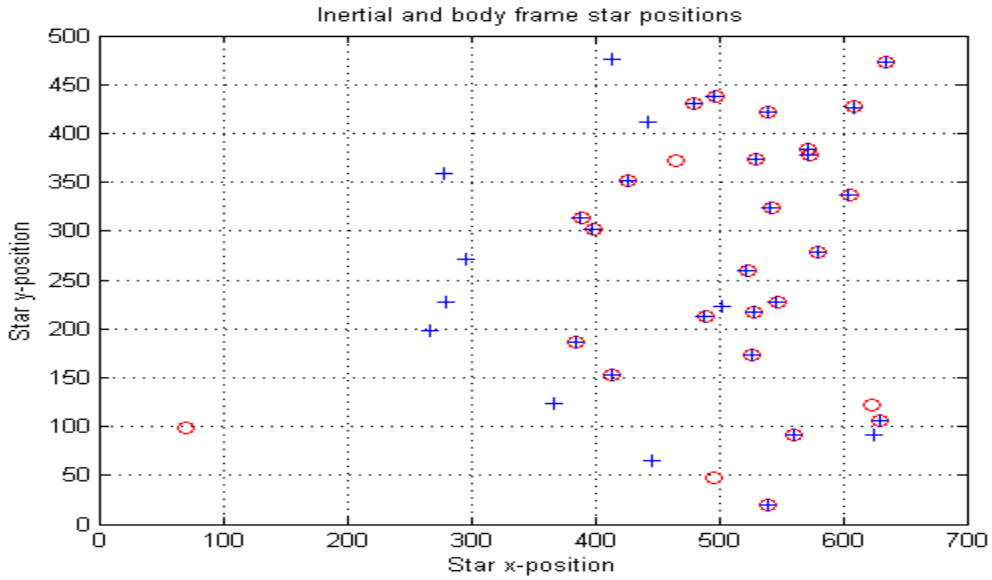

Figure 23. Planar triangle matches.

Figure 24 is the resulting image after filtering the calculations from the planar triangle algorithm. Originally, there were 24 star matches. Twenty-one matches remain after filtering using the same accuracy requirement as used in the angle algorithm. This is a significant leap, three times the amount, from the seven remaining pairs from the angle algorithm in Table 3. The additional complexity and storage requirements of the planar triangle algorithm are a significant improvement over the simplistic angle algorithm.

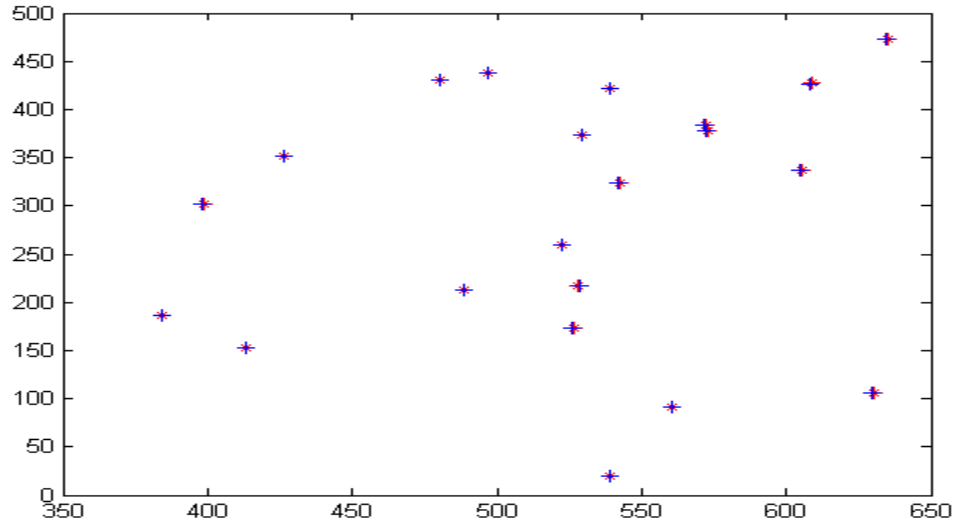

Figure 24. Filtered planar triangle matches.

c. Spherical Triangle Simulations

The last attitude determination algorithm tested was the spherical triangle algorithm. Figure 18, was used again as the database image to create a large array of areas, polar moments, and the inertial unit vectors associated with each triangle. The image in Figure 20 was used to simulate the stars viewed from an onboard star tracker. The spherical triangles were computed using Equations 8, 9, 10, 11, 12, and 13.

The same benefit of the planar triangles algorithm applies to the spherical triangles algorithm; there is no ambiguity on matching up inertial to body triangles. Both the spherical triangle area and the polar moment must match in order for the three stars of the inertial frame to be correlated to the three stars in the image spherical triangle. There will still be some error; therefore filtering is required to ensure error stars are removed.

Figure 25 represents the results of matching the inertial spherical triangles to the inertial triangles. Only seven matches are achieved with the rest being error stars. The requirements of matching both the area and polar moment of the triangle cause many misses. This is a significant miss factor for the algorithm; therefore, a significant amount of programming is required to remove those stars that are inaccurate.

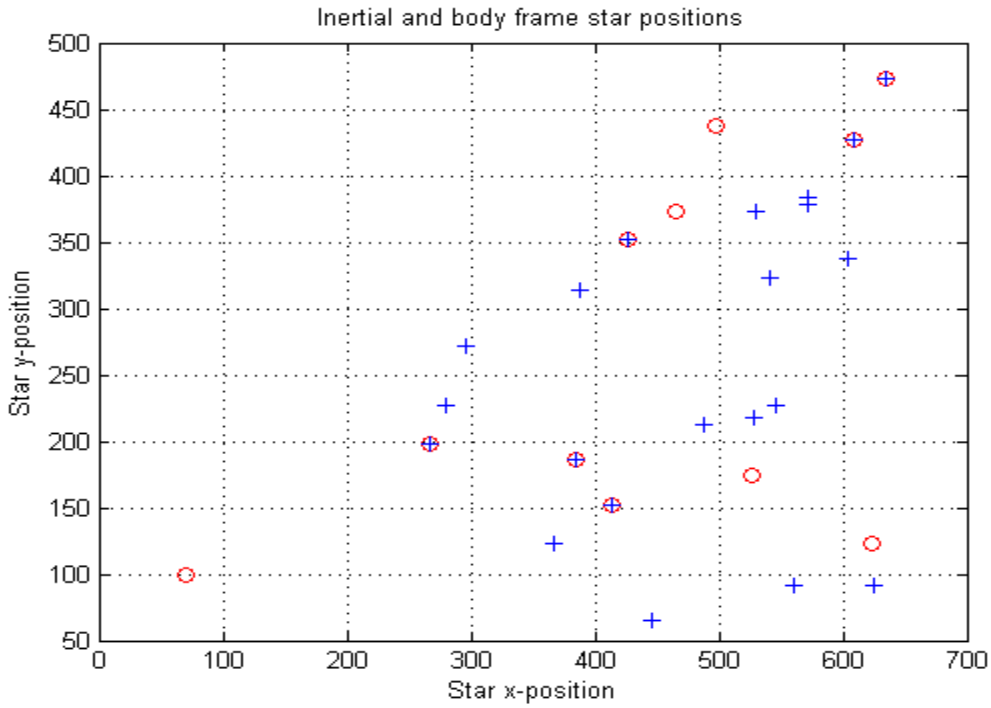

Figure 25. Spherical triangle algorithm matches.

After filtering, the final star matches can be used for attitude determination. Figure 26 is the filtered results of Figure 25. The six matches are clearly seen, and are now available for input into one of the attitude determination algorithms.

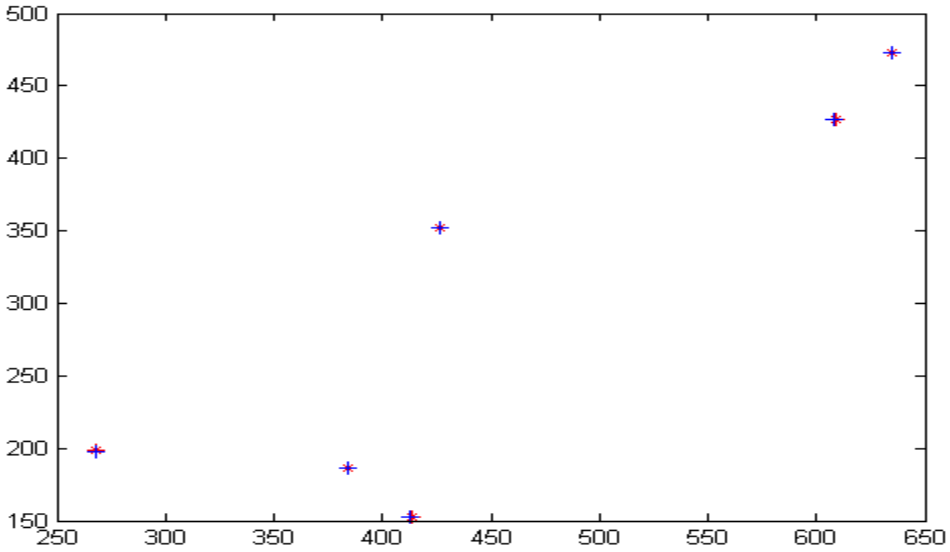

Figure 26. Filtered spherical triangle algorithm matches.

There are some pros and cons when using the spherical triangles algorithm. The spherical triangles algorithm may be comparative to the angle algorithm in performance, but does not perform nearly as well as the planar triangles algorithm. The double requirements of matching the areas and polar moments of triangles, is similar to the benefit of the planar triangles algorithm regarding ambiguity, which makes it better than the angle method. The storage requirements for spherical triangles are the same as the planar triangle. The problem comes down to the computer resources necessary to determine the polar moment.

The polar moment calculation for spherical triangles is completed by a recursive algorithm. Equation 13 is a summation of the polar moment for the triangles. As previously mentioned, the polar moment of a triangle is obtained by breaking the spherical triangle into smaller triangles. The area of each of these smaller triangles is then multiplied by the square of the arc distance from the centroid of each smaller triangle, to the centroid of the overall triangle (Cole & Crassidus 2004). The spherical triangle's polar moment is then obtained by summing the results of each smaller triangle with the process repeating until the depth of recursion is met. This is a significant

process for achieving the polar moment of a spherical triangle. The recursive algorithm for the polar moment is a significant amount of complexity on top of the other requirements for the spherical triangles algorithm with marginal returns.

3. Attitude Determination Algorithm Tests

The testing of the attitude determination algorithms was performed only on the results from the angle algorithm. The simplicity of the angle algorithm with the firm results give a background for studying attitude determination algorithms.

a. Least-Squares Testing

Equation 15 is the equation with the direction cosine matrix needed to represent the correct attitude from the inertial frame to the body frame of the satellite as shown in Figure 13. Equation 29 represents the small amounts of error or distortion in the image preventing an exact solution. Using the least squares method on the star body vectors and inertial vectors, the attitude determination matrix is found to solve Equation 28.

As described previously the least squares algorithm is a method to solve Equation 28 and obtain the direction cosine matrix when an exact solution is impossible. By putting the inertial and observed body vectors into Equation 42, and then progressively performing the mathematical manipulations of Equations 43 and 44, the result is a matrix equation that takes the form of Equation 45, which can be solved by Equation 41. In this case we want to solve for the vector $\hat{x}$ and therefore rearrange Equation 41 into Equation 46.

The results are as expected with an attitude matrix very close to the identity matrix. By performing the analysis in MATLAB the exact attitude matrix calculated is:

$$A = \begin{pmatrix} 1.0003 & 0.0004 & 0.2064 \\ 0.0007 & 0.9997 & -0.2976 \\ 0.0000 & 0.0000 & 0.9992 \end{pmatrix} \tag{67}$$

The results from the least squares algorithm are encouraging for the images taken. The direction cosine matrix does give us the attitude of the craft, but the attitude is usually represented in Euler angles. By using the transformations in Equation 27 the Euler angles for the satellite can be obtained.

b. TRIAD Algorithm Testing

Using two inertial vectors and two body frame vectors, the TRIAD algorithm was implemented for attitude determination. The vectors are identified as $\hat{V}_1$ and $\hat{V}_2$ for inertial stars with two body frame vectors from the star tracker as $\hat{W}_1$ and $\hat{W}_2$ as in Equation 47. The two triads described by Equations 48 and 49 were formed for insertion into the observed and reference matrices, M_{ref} and M_{obs} respectively, as outlined in Equation 52. The attitude matrix is obtained by Equation 53.

Using arbitrary vectors obtained from the angle algorithm, the attitude determination matrix was obtained. The attitude matrix is:

$$A = \begin{pmatrix} 1.0000 & -0.0002 & -0.0000 \\ 0.0002 & 1.0000 & 0.0000 \\ 0.0000 & -0.0000 & 1.0000 \end{pmatrix} \tag{68}$$

The matrix results are almost exactly a 3x3 identity matrix when compared to the results of the least-squares method in Equation 67.

The problem with the TRIAD algorithm as mentioned previously is that only two observed and reference vectors are used to find A. For the most accurate results from the TRIAD algorithm, the most accurate inertial and body frame vectors must be used (Shuster & Oh, 1981). In this case, it is not known which vectors are the most accurate, therefore all vectors are used. Also, portions of the second vector are discarded, therefore the algorithm is heavily influenced by whichever vector is used first (Shuster & Oh, 1981). This sensitivity for which vector is used as the first vector is extremely difficult for designers in cases where the accuracy of the vectors is unknown. For these reasons, the TRIAD algorithm should not be used for applications with high

accuracy requirments, even though it is a simplistic approach for attitude determination. In a laboratory setting, the TRIAD algorithm accuracy is sufficent .

c. QUEST Algorithm Testing

The QUEST attitude determination algorithm was the final attitude determination algorithm tested. The QUEST algorithm attempts to find a gain function that minimizes Equation 32. The same figures as the Least-Squares and TRIAD were used as the inertial database and body frame image. The vectors are identified as $\hat{V}_{1\cdots n}$ for inertial stars with the body frame vectors from the star tracker as $\hat{W}_{1\cdots n}$ for n total vector pairings.

To implement the QUEST algorithm, the complex mathematics was developed as a subroutine in MATLAB. The weighting coefficients, a_i, is determined by the number of vectors in the observations in Equation 54. The attitude profile matrix, B, was developed by summing the weighting coefficients multiplied by the vectors in Equation 55. The quantity S was calculated from adding the attitude profile matrix to its transpose as in Equation 55. The quantity σ was calculated from Equation 60 while the 3x1 vector, Z, was obtained from Equation 59. The 4x4 matrix K was formed from the previously mentioned quantities in Equation 57. Using the eig function in MATLAB, the eigenvectors and eigenvalues for the matrix K was determined. To find the optimal quaternion, the maximum eigenvalue of K was paired with its associated eigenvector.

In this case, the maximum eigenvalue has a value of one with a corresponding eigenvector of $\begin{bmatrix}0.0000 & 0.0000 & -0.0001 & 1.0000\end{bmatrix}^T$. Therefore the quaternion is:

$$q_{opt} = \begin{bmatrix}2.3619\text{e-}007 & -8.1006\text{e-}007 & -2.0437\text{e-}004 & 1.0000\end{bmatrix}^T \tag{69}$$

which correspond to:

$$q_{opt} = \begin{bmatrix}q_1 & q_2 & q_3 & q_4\end{bmatrix}^T \tag{70}$$

for this system. To determine the corresponding direction cosine matrix A is found by equation inserting the quaternions in Equation 69 into Equation 26. The attitude matrix is:

$$A = \begin{pmatrix} 1.0000 & -0.0001 & 0.0000 \\ 0.0001 & 1.0000 & 0.0000 \\ -0.0000 & -0.0000 & 1.0000 \end{pmatrix} \tag{71}$$

The attitude matrix obtained from the QUEST algorithm, which uses all the vectors is very close to the results of the TRIAD algorithm which only uses two vector pairs.

There are several benefits and drawbacks of using the QUEST algorithm for attitude determination. The complexity of the QUEST is a significant drain on computational resources especially if there are a large amount of body and inertial vectors to calculate. If there are only a few vector pairings, especially if there are only two pairings, then the TRIAD algorithm is more suitable due to its simplicity.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. TEST BED SIMULATIONS

A. ATTITUDE DETERMINATION OF THE THREE-AXIS SIMULATOR-2

The previous simulations were all conducted on images obtained from a star field with no movement of the spacecraft's body frame with respect to the ECI frame. Implementing these algorithms onto the moving Simulator-2 (TAS-2) has additional complexity because the origin locations of the frames effect the measurement of the star vectors and corresponding reference database.

1. Equipment

a. Three-axis Simulator 2 (TAS-2)

The TAS-2 used for the experiment is a second-generation three-axis spacecraft simulator used in the Spacecraft Research Design Center (SRDC) at the Naval Postgraduate School. The TAS-2 was designed and built to act as a test bed for the development and validation of acquisition, tracking, and pointing techniques. The five-foot diameter simulator floats on a spherical air bearing to allow it to rotate freely and simulate space flight. The positioning of the simulator is accomplished by three, variable-speed, control moment gyroscopes.

The TAS-2 simulates spacecraft three-axis motion and has a payload of an optical system, which simulates two gimbaled space telescopes along with multiple attitude sensors. The sensors already installed on the platform are Inertial Measurement Units (IMUs) with integrated rate gyros, for angular rate, a two-axis analog sun sensor, and two inclinometers. Added to these instruments is a camera to simulate a star tracker.

The TAS-2 also has several onboard computers installed, one of which is used for conducting experiments associated with this project. The image capturing and star-centroiding software has been saved to the computer for processing the star fields and obtaining the necessary information for the algorithms. All algorithms associated with testing of the star tracker are also stored on this computer.

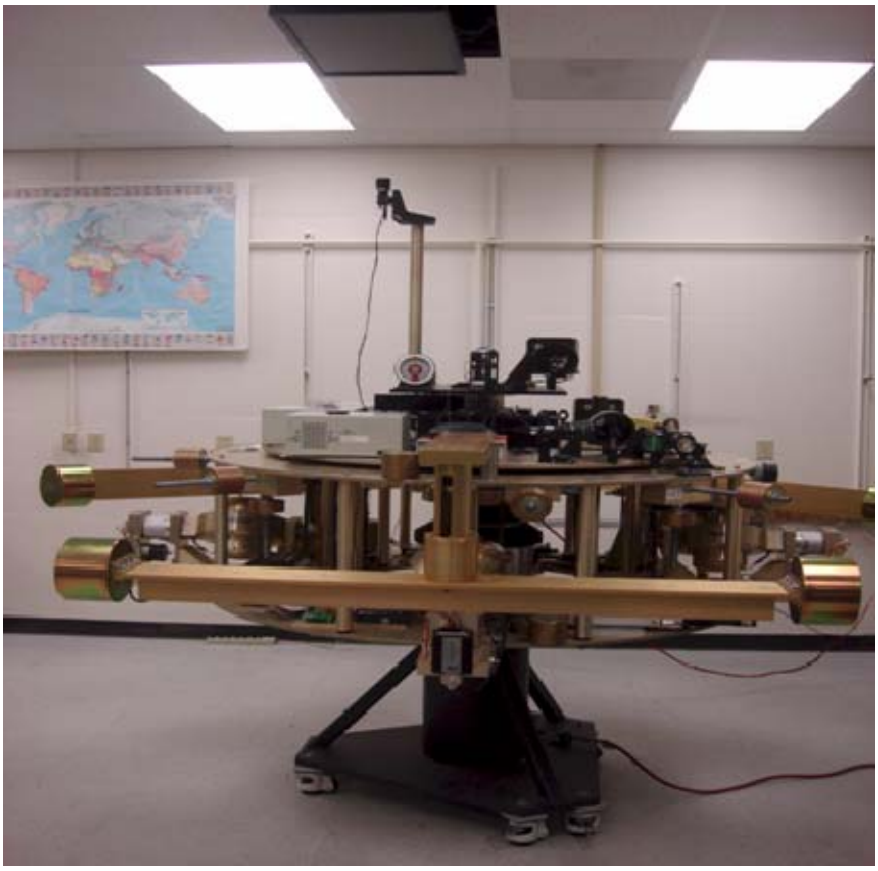

Figure 27. TAS-2 with star field monitor installed above it.

b. Star Field Image

Another star field was created for use with the TAS-2 three-axis simulator. This star field was a generic star field intended only for attitude determination and is given as Figure 28. The field was projected on the screen above the TAS-2. The star field is setup to minimize stars close to each other for algorithm testing. The star field is easily picked up by the camera when the room is darkened.

Figure 28. Star field for TAS-2 testing.

c. Star Tracker Camera

A new Pentax wide FOV lens with an 8.5 mm focal length was installed to allow a wider angle of detection of the entire star field for use in the algorithms. The camera and lens was installed such that all the stars are visible to the camera for a small range of motion. Figure 29 shows the upgraded lens on the camera, the additional standoff on top of the three-axis simulator to get the camera closer to the star field, as well as the screen with the field in the darkened lab.

Figure 29. Camera and screen for TAS-2 testing.

The star field images are again processed using MATLAB, however many pixels were picked up where there was no star. There are therefore several hot pixels (stuck pixels) on the CCD, or pixels with higher than normal rates of charge leakage. By placing a cap over the lens, and performing the image capturing routine in MATLAB, an image with all the hot pixels has been obtained. This background image is then subtracted from all other images when conducting tests to remove any hot pixels preventing them from becoming error stars.

B. ATTITUDE DETERMINATION OF THE TAS-2

1. Algorithm Choices

The first step is determining which star pattern recognition algorithm should be used for the inertial database of the TAS-2. Since this is the first implementation of a star tracker and the algorithms associated with it, the angle algorithm was chosen.

The attitude determination algorithm used for testing on the TAS-2 is the QUEST algorithm. The use of the stars in the FOV and simple eigenvalue calculations on the 4x4 K matrix make the algorithm suitable for testing. The QUEST algorithm has been modified to return the optimal quaternion as well as the attitude determination matrix.

In previous simulations, the star vector measured at zero attitude is considered as a reference database. As seen in Figure 29, the star tracker camera is approximately one meter below the screen with the star field. The TAS-2 is a floating simulator that can perform roll, pitch, and yaw maneuvers. Since the star tracker camera cannot be located at the center of rotation of the simulator the frame fixed to the star tracker will not only be rotated but also translated. In space, this translational motion of the frame is not a problem since the offset of the camera frame and the satellite's body frame to the inertial frame does not affect angles or triangles of stars since the stars are extremely far away. In a laboratory environment, this translational motion will affect the star pattern recognition algorithm.

2. Creating an Inertial Database

The first step is to develop an inertial database for the TAS-2. With the TAS-2 completely level, 200 images are taken of the ceiling-positioned star field. A total of nine stars are displayed on the LCD screen for the experiment. For each of the 200 images, the background noise is subtracted. The averages of the x and y positions will be taken over the 200 images to create a final, averaged image. This image is used to create an inertial database for all test-bed simulations. Since the image is taken in the camera frame, it must be shifted to an inertial frame of reference. The inertial coordinate system is centered at the center of rotation of the TAS-2 which is at the center of the spherical air bearing.

Figure 30 illustrates the star field, the inertial frame I centered at the center of rotation of the spacecraft simulator, the camera fixed frame B, the translated inertial frame I' and the translated body frame B'. The frame I is the non-moving inertial frame for the system. The center of the frame I' moves with the camera, but does not rotate and its axes are parallel to those of frame I. The translation of frame I from frame I' is done by position vector $\tilde{R}_i$. Frame B is the frame associated with the camera and therefore is fixed with the camera. The frame B' is translated from frame B such that the origin of frame B' coincides with the origin of the inertial frame I.

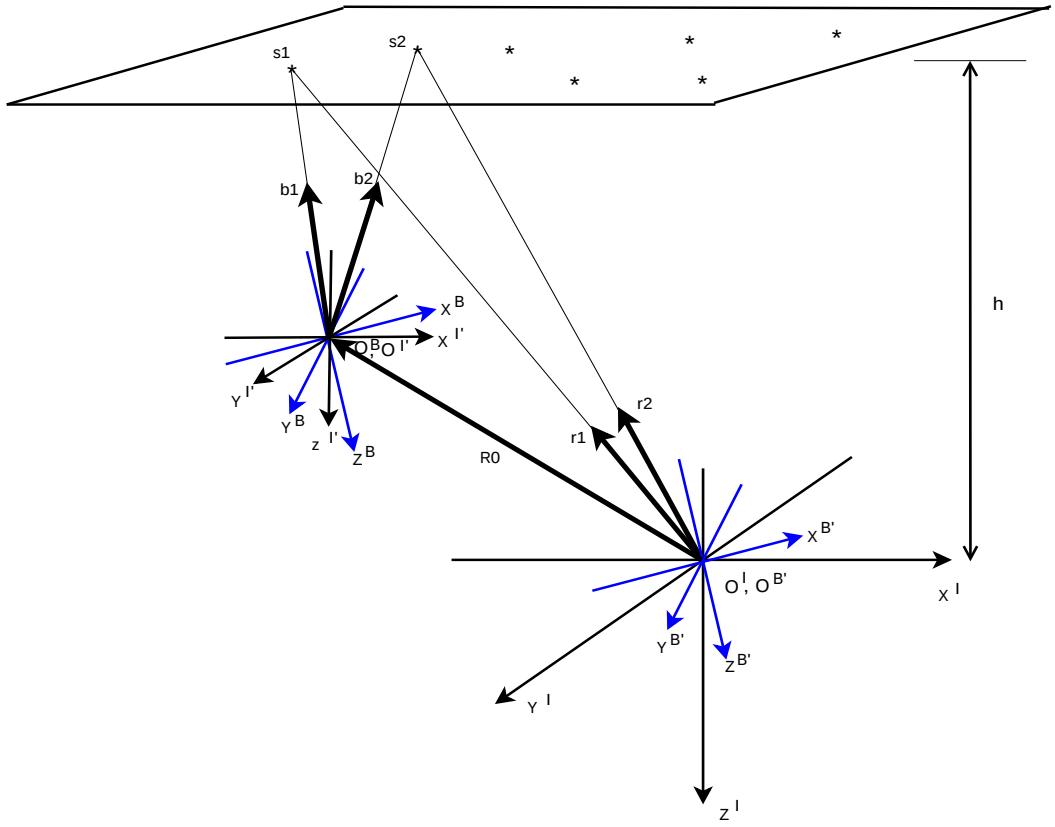

Figure 30. Vector representations of star field, star tracker frame, and the inertial frame.

For star fields located at a far distance, the unit vectors $\tilde{b}_i$ represented in frame B are related to the unit star vector $\hat{r}_i$ represented in frame I by the following relationship:

$$b_i^B = Ar_i^I \tag{72}$$

where A is the direction cosine matrix representing the attitude of the spacecraft simulator and the superscripts denote the specific frame used for the star vectors. In addition, the angle between b_1 and b_2 is same as the angle between r_1 and r_2 for distant stars. When the stars are displayed in close proximity, the angle between b_1 and b_2 is not same as the angle between r_1 and r_2 as can seen in Figure 30.

In order to solve this problem, let us first define α_i representing the distance between ith star and the origin of frame B (or frame I'). Similarly, we can also define β_i

being the distance between ith star and the origin of frame *I* (or frame B'). Using α_i and β_i, the following relationship can be found.

$$R_0 + \alpha_i b_i = \beta_i r_i \tag{73}$$

Equation 73 can be rewritten as follows

$$R_0^{B'} + \alpha_i b_i^B = A\beta_i r_i^I \tag{74}$$

From Equation 74, $R_0^{B'}$ is a constant vector fixed to the spacecraft body, b_i^B is measured by the star tracker camera, and r_i^I is a star vector represented in the inertial frame which will serve as the database. Equation 74 is not a linear equation to solve for the attitude matrix because α_i is a function of *A*.

Assuming that the star tracker camera is looking at the $-z^B$ direction and defined a vector $p^{I'} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^{\mathrm{T}}$. The distance from the origin of the inertial coordinate system to the monitor is 1.7695 m which is defined as *h* in Figure 30. The α is derived from:

$$\alpha_i \left(\hat{b}_i^{I'} \bullet \hat{p}^{I'} \right) = h - \hat{p}^{I'} \tilde{r}^I \tag{75}$$

Equation 75 can be rearranged so that:

$$\alpha_i = \frac{h - \left(\hat{p}^{I'} \right)^T A^T \tilde{r}_o}{\left(\hat{p}^{I'} \right)^T A^T \tilde{b}_i^B} \tag{76}$$

The β_i can also be computed as

$$\beta_i = \frac{h}{\left(p^{I'} \right)^T r_i^I} \tag{77}$$

For reference database, the star vectors measured in frame *B* at zero attitude need to be converted into star vectors in frame I. From Equation 74, the unit star vector at the inertial frame *I* which will serve as a database becomes

$$r_i = \frac{R_o{}^{B'} + \alpha_i b_i^B}{\beta_i} \tag{78}$$

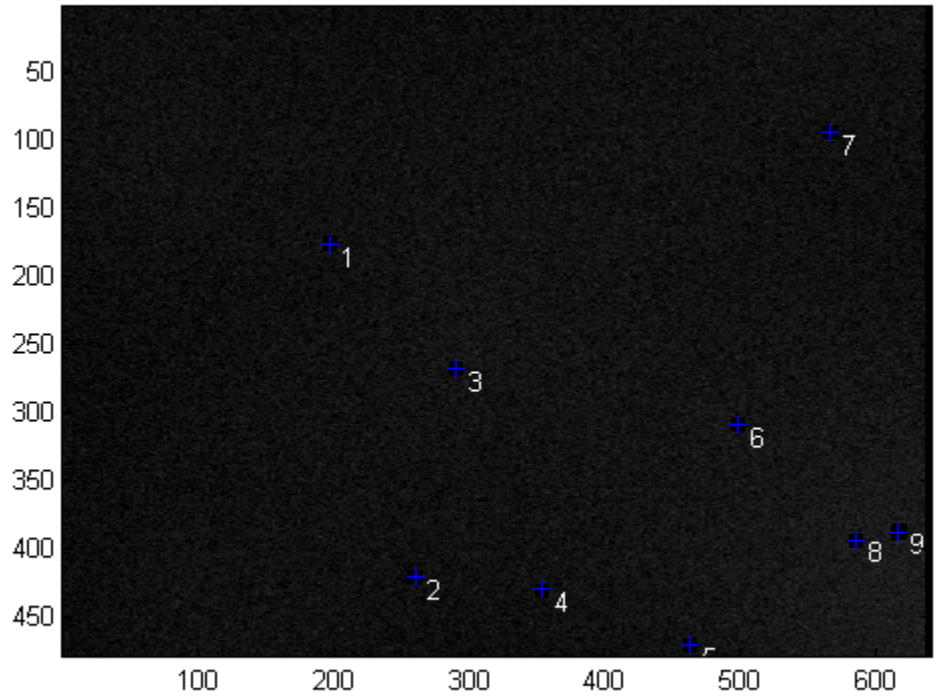

Figure 31. Inertial star database image with numbering.

Figure 31 illustrates an image from the database stars with the respective star numbers. Once, the vectors are obtained in inertial frame *I*, the angle database is completed. Using Equation 2, all the inertial angles are calculated. The inertial numbers of the stars used to calculate the angles are stored with their respective angles to create a lookup table. The entire database is stored as a MAT file in the TAS-2 computer system. The final database comprises angles with star numbers and the table with the inertial unit vectors and star number.

3. TAS-2 Testing Iterations

The image capturing and centroiding software in MATLAB is again used to detect the centroids of the stars in the image for calculating the unit vectors in the camera frame B.

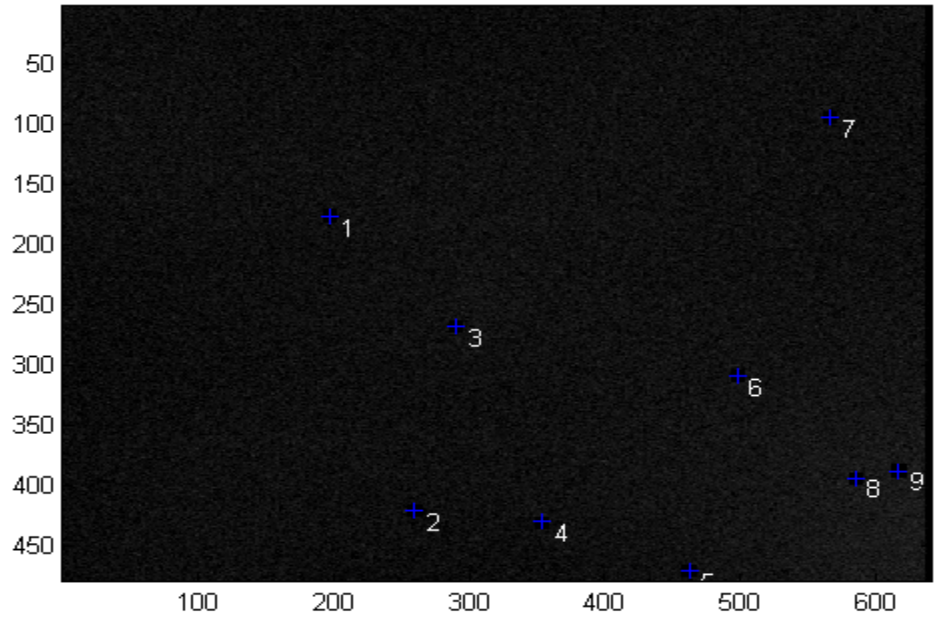

Figure 32. Stars observed by TAS-2 star tracker.

The stars are picked up by the star tracker camera in Figure 32. The stars are arranged so that the master star is the star closest to the y-axis with successive stars numbered according to their proximity to the y-axis. These stars have their unit vectors calculated in the *B* frame. There are nine stars counted, therefore there are eight angles between the master star and all the eight other stars in the image. Again, the master star in an image is the star with the highest intensity.

Equation 74, $R_0^{B'} + \alpha_i b_i^B = A\beta_i r_i^I$, is not easily applicable for computation of attitude matrix *A* because α_i is also a function of *A*. Instead of solving complex optimization problem, we want to apply the algorithms presented and verified with the computer simulation in the previous chapter. In order for this, a simple iterative approach is proposed. First, compute the vectors $R_0^{B'} + \alpha_i b_i^B$ using the prediction of a spacecraft attitude *A*. The prediction of a spacecraft attitude can be made from previous estimates of the spacecraft attitude or using additional sensors such as rate gyros. With this attitude

prediction, the sect of vectors, $R_0^{B'} + \alpha_i b_i^B$, can be used to compute the angles between them. These angles are now compared with the angles in the database and set of matching stars are identified for attitude estimation. If the prediction of the attitude is not accurate, the accuracy for the angle matches need to be relaxed. In order for accurate estimation, the resulting attitude estimation can again be used to perform more accurate matches and attitude estimation. Therefore, this method needs several iterations with slight increases in computational power.

Several testing iterations are planned to test the fidelity of the angle algorithm and the QUEST attitude determination algorithm for the test-bed. For the experiment, star unit vectors translated to the B' frame from the star image are first computed, and the angles between the brightest or master star and all other stars are calculated. These angles are compared to the inertial angles stored in the database.

The testing program is setup so that the prediction of *A* attitude matrix used for computation of $R_0^{B'} + \alpha_i b_i^B$ has either no errors or some errors while the test-bed is at zero attitude. By doing these tests, we can measure the accuracy of the estimation as well as the required accuracy of the prediction of the *A* matrix for the proposed algorithm.

A series of *A* matrices with an error of six degrees, three degrees, zero degrees, minus three degrees, and minus six degrees were chosen. An accuracy of 500 arc-seconds for each angle was chosen for all *A* matrices. Therefore, an angle from an observed star angle must match an inertial angle by the value ± 250 arc-seconds. All the multiple matches were filtered out to ensure accurate results in the experiment.

After the angles are matched, the star inertial vectors and observed star vectors are then entered into the QUEST algorithm. The QUEST algorithm will then calculate an updated or accurate *A* matrix and attitude quaternion. This new *A* matrix can then be used as a new initial estimate for further attitude calculations.

For each test with 50 attitude estimations, several parameters were observed. The resulting average updated *A* matrix from the QUEST algorithm for each test with its

standard deviation were recorded. The number of stars and angles matched were saved, as well as the Euler angles and their standard deviations. For these tests, there is only one iteration of attitude updates.

a. Testing of an A of Zero Degree Error

With a prediction of the *A* matrix of zero degree error, in other words a 3x3 identity matrix, five test runs with 50 attitude determinations for each test were accomplished with a 500 arc-second accuracy. The results of the attitude testing for the Euler angles and their standard deviations are included in Table 4. Even though there is no error inserted, only seven stars out of nine are easily matched, due to the noise of the system. The mean values and standard deviations for all the Euler angles remain fairly constant over the five tests as well.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	6.86E-04	5.65E-04	6-004	5.65E-04	9.28E-04	6.86E-04
Theta	3.70E-03	3.00E-03	2.00E-03	3.00E-03	5.00E-03	3.34E-03
Psi	2.97E-05	2.30E-04	1.30E-04	2.30E-04	4.30E-04	2.10E-04
σ Phi	1.20E-03	1.10E-03	9.18E-04	1.10E-03	1.40E-03	1.14E-03
σ Theta	0.0066	0.0061	0.005	0.0061	0.0074	6.24E-03
σ Psi	6.74E-04	6.18E-04	5.06E-04	6.18E-04	7.57E-04	6.34E-04
Number of stars matched	7.000	7.000	7.000	7.000	7.000	7.000

Table 4. Tabulated results for Euler angles and standard deviation for an *A* matrix of zero degree error.

The average attitude matrix for this testing is:

$$A = \begin{pmatrix} 0.9943 & 0.0002 & 0.0027 \\ 0.0003 & 0.9927 & 0.0006 \\ 0.0033 & 0.0005 & 0.9983 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0121 & 0.0006 & 0.0065 \\ 0.0006 & 0.0154 & 0.0011 \\ 0.0062 & 0.0012 & 0.0035 \end{pmatrix}$$

As seen from these simulations, the attitude is approximately equal to the true attitude (identity matrix) due to no errors in the predicted attitude. There is very little standard deviation of the *A* matrix and most of the stars are picked up by the star tracker. The next phase of testing is with an error in the predictive attitude.

b. Testing of an A With Three Degrees Error

The next phase of testing inserted an error of 3 degrees into the Euler angles to form an initial *A* matrix used in attitude determination. Five more test runs for attitude determinations were accomplished with a 500 arc-second accuracy.

The results of the attitude testing for the Euler angles and their standard deviations are included in Table 6. The QUEST and angle algorithms do correct for the error, but fewer stars are matched. The main thing noticeable from this test is the marked decline in the number of stars detected by the algorithm. Only 3.26 stars are detected, due to the initial error during 50 runs. If the initial attitude estimate is off, the results from the angle algorithm decline sharply.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	1.70E-03	6.58E-04	2.90E-03	3.50E-03	2.70E-03	2.29E-03
Theta	7.40E-03	1.70E-03	1.36E-02	1.71E-02	1.26E-02	1.05E-02
Psi	3.30E-04	-2.41E-04	9.58E-04	1.30E-03	8.54E-04	6.40E-04
σ Phi	3.70E-03	1.58E-04	5.50E-03	6.20E-03	5.40E-03	4.19E-03
σ Theta	0.0199	4.18E-04	0.0299	0.0335	0.0294	2.26E-02
σ Psi	2.10E-03	5.77E-05	3.10E-03	3.40E-03	3.00E-03	2.33E-03
Number of stars matched	3.260	3.320	3.200	3.320	3.200	3.260

Table 5. Tabulated results for Euler angles and standard deviation for an *A* matrix of three degree error.

The average attitude matrix for this testing with a three degree error is:

$$A = \begin{pmatrix} 0.9832 & 0.0007 & 0.0073 \\ 0.0012 & 0.9786 & 0.0023 \\ 0.0131 & 0.0020 & 0.8006 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0445 & 0.0139 & 0.1072 \\ 0.0144 & 0.0556 & 0.0357 \\ 0.1205 & 0.0896 & 0.0188 \end{pmatrix}$$

c. Testing of an A With a Six-Degree Error

Further testing with a large error of six degrees was attempted with an accuracy of 500 arc-seconds. This error would test the ability of the algorithms to arrive at the correct attitude solution with a large initial estimate error.

The results of this test were poor. To get the algorithm to function, the accuracy had to be dropped to 900 arc-seconds or 0.0044 radians to get two angles to match with the database correctly. Four angles were detected, but only two were accurate. The only way to make the algorithm work with this amount of error, is to filter the database further for angles that are within 0.0044 radians of each other.

d. Testing of an A With a Minus Three-Degree Error

The next series of tests involved using an A matrix with an Euler error of minus three degrees. The error was inserted and the simulations ran. Table 7 contains the results of the minus three degree angle error tests. The accuracy is maintained at 500 arc-seconds for all tests.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	-3.71E-04	-2.92E-04	-5.49E-05	-2.91E-04	-7.65E-04	4.30E-03
Theta	7.51E-04	5.86E-04	9.27E-05	5.85E-04	1.60E-03	1.06E-02
Psi	-8.86E-05	-7.20E-05	-2.26E-05	-7.23E-05	-1.71E-04	1.20E-03
σ Phi	1.20E-03	1.10E-03	5.59E-04	1.10E-03	1.60E-03	9.10E-03
σ Theta	0.0025	0.0023	0.0012	0.0023	0.0033	2.48E-02
σ Psi	2.50E-04	2.26E-04	1.17E-04	2.26E-04	3.33E-04	3.50E-03
Number of stars matched	6.020	6.000	6.000	6.000	6.000	6.004

Table 6. Tabulated results for Euler angles and standard deviation for an A matrix with minus three degrees error.

From the results in Table 6, the Euler Angles are accurately calculated, but the star matches drops from the nine matches achieved with a zero radian error. Only six of the stars are matched in most of the cases with an initial error.

The average attitude matrix for this testing with a minus three-degree error is:

$$A = \begin{pmatrix} 0.9939 & -0.0001 & 0.0014 \\ -0.0001 & 0.9937 & -0.0006 \\ 0.0012 & -0.0006 & 0.9997 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0196 & 0.0004 & 0.0040 \\ 0.0004 & 0.0202 & 0.0020 \\ 0.0041 & 0.0019 & 0.0011 \end{pmatrix}$$

e. Testing of an A With a Minus Six-Degree Error

With an *A* matrix of minus six degrees of error, five test runs for attitude determination were accomplished with a 500 arc-second accuracy. The results of the

attitude testing for the Euler angles and their standard deviations are included in Table 7. The error still allows six stars to match in all cases.

	Run 1	Run 2	Run 3	Run 4	Run 5	Overall Average
Phi	-6.86E-04	-4.49E-04	-6.08E-04	-4.49E-04	-6.07E-04	4.30E-03
Theta	1.40E-03	9.12E-04	1.20E-03	9.13E-04	1.20E-03	1.06E-02
Psi	-1.55E-04	-7.20E-05	-1.38E-04	-1.05E-04	-1.38E-04	1.20E-03
σ Phi	1.50E-03	1.30E-03	1.50E-03	1.30E-03	1.50E-03	9.10E-03
σ Theta	0.0032	0.0027	0.003	0.0027	0.0015	2.48E-02
σ Psi	3.20E-04	2.71E-04	3.06E-04	2.71E-04	3.06E-04	3.50E-03
Number of stars matched	6.000	6.000	6.000	6.000	6.000	6.000

Table 7. Tabulated results for Euler angles and standard deviation for an A matrix with minus six degrees error.

The average attitude matrix for this testing with a minus six degree error is:

$$A = \begin{pmatrix} 0.9939 & -0.0001 & 0.0014 \\ -0.0001 & 0.9937 & -0.0006 \\ 0.0012 & -0.0006 & 0.9997 \end{pmatrix}$$

while the standard deviation for this matrix is:

$$\sigma = \begin{pmatrix} 0.0125 & 0.0003 & 0.0026 \\ 0.0003 & 0.0130 & 0.0013 \\ 0.0026 & 0.0012 & 0.0007 \end{pmatrix}$$

5. TAS-2 Testing Iterations with Attitude Updates

The next testing incorporated using the QUEST algorithm to provide an updated attitude matrix to assist with the attitude solution. The updated attitude solution should

increase the accuracy by providing by providing corrected *A* matrices as initial estimate to the algorithms matching the inertial database to the body frame image.

The update testing was completed with an initial error of two degrees in the TAS-2 initial attitude estimate with an accuracy requirement of 500 arc-seconds. The testing starts at a single update, and then continues on up to five attitude updates. Theoretically, at each update, the attitude solution will improve. Table 8 contains the results of the testing.

	update 1	Update 2	Update 3	Update 4	Update 5	Overall Average
Phi	1.30E-03	7.98E-04	1.50E-03	2.40E-03	1.30E-03	4.30E-03
Theta	6.50E-03	3.20E-03	4.00E-03	3.10E-03	4.00E-03	1.06E-02
Psi	5.05E-04	3.13E-04	-1.33E-04	-1.63E-04	-6.44E+00	1.20E-03
σ Phi	2.30E-03	1.70E-03	2.90E-03	3.40E-03	2.90E-03	9.10E-03
σ Theta	0.0125	0.0094	0.0087	0.0073	0.0093	2.48E-02
σ Psi	1.30E-03	9.91E-04	1.10E-03	1.30E-03	1.10E-03	3.50E-03
Number of stars matched	5.000	6.640	6.680	6.600	6.630	6.310

Table 8. Tabulated results for Euler angles and standard deviation with *A* matrix updates.

The main result of the update test is the increasing accuracy of the initial estimate of the attitude solution. With an initial error of two degrees, the updates remove the error and provide an updated attitude matrix. As shown in the bottom line, the attitude updates improve with increasing amounts of attitude solution updates. The amount of star matches improves from 5.0 at the beginning to ~6.6 stars at the end with five updates. Table 9 shows the increase in star identification with increasing updates, while Table 10 shows the Euler angles over the same updates.

By using an updated attitude matrix with errors removed while holding accuracy constant, star matching improves rapidly. With only one update, only five star matches

are achieved. When using two updates, the matches increase to almost seven stars identified, which is very close to the testing in Table 4, which was conducted using an *A* matrix of zero error.

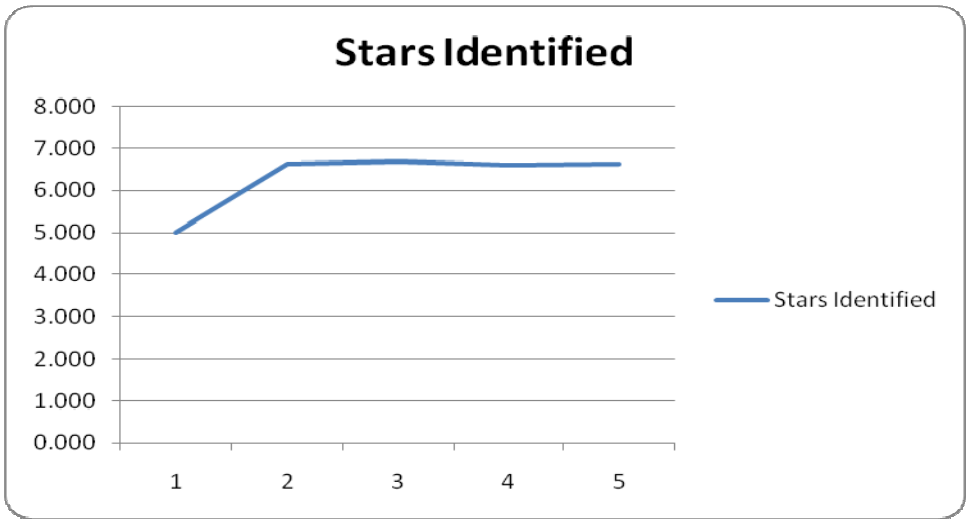

Table 9. Tabulated results for Stars Recognized and standard deviation for an A matrix with a two degree error.

With increasing *A* matrix updates, more precise Euler angles should also be obtained from the more accurate *A* matrix. The Euler angles, shown in Table 10, overall exhibit similar trends of increasing accuracy. Most improved are the Theta and Psi angles

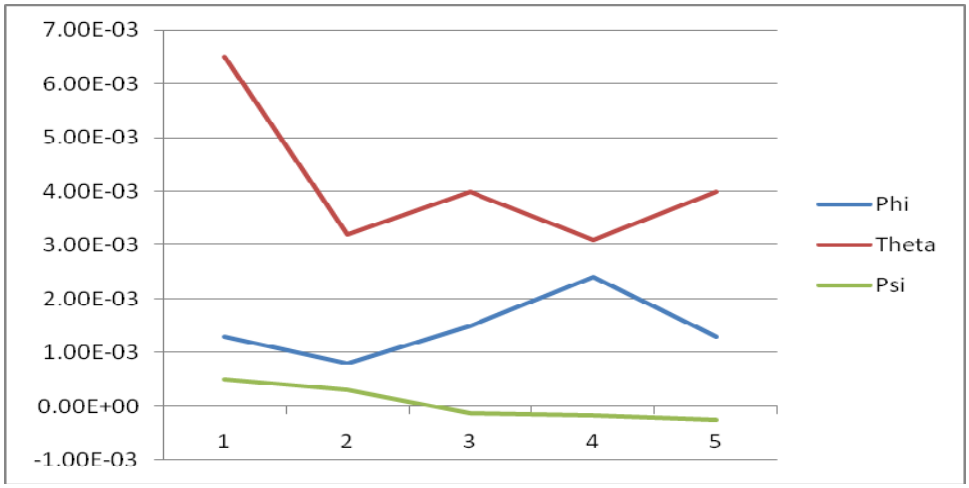

Table 10. Phi, Theta, and Psi angles plot over iterations.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

A. STAR PATTERN RECOGNITION ALGORITHM SIMULATIONS

The three star pattern recognition algorithms studied were the angle algorithm, the planar triangle algorithm, and the spherical triangle algorithm. Each algorithm was simulated using the same images for an inertial database and for a star image as seen from on orbiting spacecraft. After review, the algorithm with the best performance based on computational resources, storage requirements, and performance was chosen.

The computational requirements and computer resources for the algorithms varies with each algorithm. The angle algorithm calculation is very simplistic and requires very little computational power. Both the triangle algorithms require far more calculations than the angle algorithm. The added complexity of the triangle algorithms is offset by the additional logic necessary for determining which star is which in the angle algorithm. The hardest algorithm for computational requirements is by far the spherical triangle algorithm with the recursive calculations of Equation 13 to determine polar moment.

Each algorithm has its own specific storage requirements. The angle algorithm is simplistic requiring two star unit vectors to be stored with each angle. The planar triangle algorithm is more complex since there are two data points, polar moment and area, which must be stored for each triangle. Additionally for the planar triangles moment and area, three star unit vectors must be stored with each of them. The same is true with the spherical triangle algorithm.

The algorithm that performs the best regarding computational requirements, results, and storage requirements is the planar triangle algorithm. The accuracy of the algorithm regarding matches beats the angle and spherical triangle algorithms. The storage requirements are more than the angle algorithm and the same as the spherical triangle, but the amount of accurate hits more than offsets the angle algorithm. The planar triangle is comparative to the angle algorithm regarding computing resources but beats the spherical triangle algorithm.

B. ATTITUDE DETERMINATION ALGORITHM SIMULATIONS

There are many attitude determination algorithms also in use to solve what is known as Wahba's problem, which was introduced in this report. Three attitude determination algorithms have been simulated and studied, using the results from the angle algorithm. Each algorithm has its specialties and drawbacks. The TRIAD and QUEST algorithms are extremely accurate differing only slightly than the least-squares algorithm. The QUEST and least-squares use all the stars whereas the TRIAD algorithm is only used on two stars with the remaining stars in the FOV being discarded.

The QUEST algorithm is the most suitable algorithm for purposes of this project. All stars are used in the algorithm and the results are very accurate. Even though the algorithm involves complex mathematics, the eigenvalue computations are far less intensive than the least-squares algorithm. Specifically of benefit, is that no matter how many stars are used in determining the solution, the result is always obtained by finding the eigenvalues and eigenvectors of the matrix K in Equation 57. The K matrix will always be a 4x4 matrix as well which reduces the computational resources.

The TRIAD and least-squares algorithms achieve the solutions, but they also have many drawbacks. The TRIAD algorithm is simplistic, but it involves only two observed vectors and two inertial vectors. The accuracy of the results are based solely on the accuracy of the chosen vectors. If two vectors are inaccurate, the entire solution is in error, even though more accurate vectors cannot be used since the algorithm is designed for only two. On top of that, the TRIAD algorithm is heavily influenced by the vector that is chosen first. For these reasons, the QUEST algorithm is the algorithm of choice for attitude determination.

C. TAS-2 IMPLEMENTATIONS

The angle algorithm for star pattern recognition and the QUEST algorithm for attitude determination were successfully implemented on the TAS-2. The algorithm worked with an accuracy of 500 arc-seconds with an initial estimate of position as a 3x3 identity matrix. The algorithms accurately determined the TAS-2 position for a range of

error from three degrees to minus six degrees. Beyond three degrees and minus six degrees the angle method breaks down in its ability to accurately determine the position of the TAS-2.

By updating the A matrix by outputs from the QUEST algorithm, the accuracy of the attitude solution increased markedly until a certain point, then leveled off. The maximum star recognition increased from approximately five stars with one update and up to 6.6 stars with five updates, then levels off. The testing shows that increased accuracy is obtained by providing an updated attitude solution as an initial estimate to the algorithms.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. FUTURE WORK

A. FUTURE WORK ON THE TAS-2 AND STAR TRACKERS

There is still much work to be done on the TAS-2 and with star tracker algorithms and attitude determination algorithms. There are many other types of attitude determination and star pattern recognition algorithms to study and implement. Other work involves incorporating other sensors into the attitude determination problem.

Further study in algorithms should also take place. The planar triangle and spherical triangle star pattern recognition algorithms need to be done as well as an Optimized TRIAD algorithm for attitude determination.

The next step regarding instruments on the TAS-2 is routing the IMU data to the attitude determination algorithms to provide an initial estimate of the TAS-2. By receiving IMU data, increase accuracy and steering the TAS-2 may be possible.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Agency, E. S. (2009, October 16). *Hipparcos.* Retrieved October 16, 2009, from ESA Space Science: http://www.esa.int/easSC/120366_index_0_m.html

Bar-Itzack, Itzhack Y; Harman, Richard R. Optimized TRIAD Algorithm For Attitude Determination. Flight Dynamics Support Branch, Code 553. Greenbelt, MD.

Christian Liebe, C. (2002). Accuracy Performance of Star Trackers—A Tutorial. *IEEE Transactions On Aerospace And Electronic Systems*, 587–599.

Cole, C. C. (2004). Fast Star Pattern Recognition Using Planar Triangles. *AIAA/AAS* .

Cole, C. L., & Crassidus, J. L. (2004). Fast Star Pattern Recognition Using Spherical Triangles. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit.* Providence, Rhode Island: AIAA.

Crossbow Technology, Inc. (2009, November 27). *www.xbow.com.* Retrieved November 27, 2009, from Crossbow: www.xbow.com

Curtis, H. D. (2005). *Orbital Mechanics for Engineering Students.* Norfolk, Great Britain: Elsevier Buterworth-Heinemann.

Diaz, K. D. (2006). *Performance Analysis Of A Fixed Point Star Tracker Algorithm.* San Luis Obispo: California Polytechnic State University.

Gwanghyeok, J., & Junkins, J. L. (2003). Overview Of Star Tracker Technology And Its Trends In Research And Development. *American Astronomical Society*, 461–477.

Liebe, C. C., Dennison, E. W., Hancock, B., Stribl, R. C., & Pain, B. *Active Pixel Sensor (APS) based Star Tracker.* Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.

The Mathworks, (2009, November 27). *www.mathworks.com.* Retrieved November 10, 2009, from http://www.mathworks.com/access/helpdesk/help/toolbox/aeroblks/f3-22568.html.

Needelman, D. D., Li, R., & Wu, Y.-W. A. (2005). Recent Advances In Stellar Attitude Acquisition (SAA) Algorithms And Procedures. *AIAA Guidance, Navigation, and Control Conference and Exhibit* (pp. 1–17). San Francisco: AIAA.

Shuster, M., & Oh, S. (1981). Three-Axis Attitude Determination From Vector Observations. *Journal of Guidance and Control*, 70–77.

Spratling, B. B., & Mortari, D. (2009). *A Survey On Star Identification Algorithms.* College Station: Texas A&M University.

The Aerospace Press. (2002). *Spacecraft Thermal Control Handbook. Volume I: Fundamental Technologies. Second Edition.* El Segundo, California: American Institute of Aeronautics and Astronautics, Inc. and The Aerospace Press.

Wahba, G. (1966). Problem 65-1, A Least Squares Estimate of Satellite Attitude. *Society for Industrial and Applied Mathematics*, 385–386.

Weiss, C. H., Bar-Itzhack, I. Y., & Oshman, Y. (2005). Quaternion Estimation From Vector Observations Using a Matrix Kalmann Filter. *AIAA Guidance, Navigation, and Control Conference and Exhibit.* San Francisco, CA: AIAA.

Wie, B. (1998). *Space Vehicle Dynamics and Control.* Reston, VA: American Institute of Aeronautics and Astronautics, Inc.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California